

ERKENNUNG KRITISCHER ZUSTÄNDE EXOTHERMER CHEMISCHER REAKTIONEN DURCH EXPERTENSYSTEME

**Grundlagen des Entwurfs,
Restriktionen der Wissensmodellierung
und Anwendung wissensbasierter Methoden
auf ein technisches Problem**

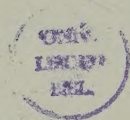
DISSERTATION
zur Erlangung des Grades eines
Doktor-Ingenieurs

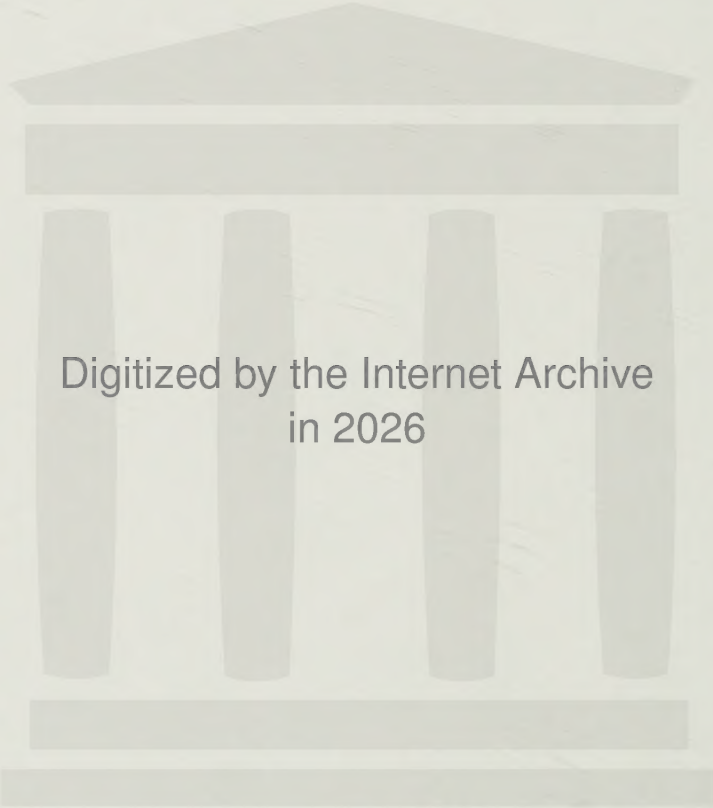
dem Fachbereich Elektrotechnik/Physik
der Universität Bremen

vorgelegt von

Dipl.-Ing. U. Meyer

Bremen 1988





Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41543102_0008

ERKENNUNG KRITISCHER ZUSTÄNDE EXOTHERMER CHEMISCHER
REAKTIONEN DURCH EXPERTENSYSTEME

Grundlagen des Entwurfs, Restriktionen
der Wissensmodellierung und Anwendung
wissensbasierter Methoden
auf ein technisches Problem

DISSERTATION
zur Erlangung des Grades eines
Doktor-Ingenieurs

dem Fachbereich Elektrotechnik/Physik
der Universität Bremen
vorgelegt von

Dipl.-Ing. U. Meyer

Bremen 1988

Diese Arbeit hat dem Promotionsausschuß Dr.-Ing. der
Universität Bremen unter dem Titel

"Erkennung kritischer Zustände exothermer chemischer Reaktionen
durch Expertensysteme - Grundlagen des Entwurfs, Restriktionen
der Wissensmodellierung und Anwendung wissensbasierter Methoden
auf ein technisches Problem"

vorgelegen.

An dem Promotionsverfahren haben Prof. Dr.-Ing. D. Popovic und
Prof. Dr.-Ing. W. Sobotta als Gutachter mitgewirkt.

Das Kolloquium hat am 3. Juni 1988 stattgefunden.

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Automatisierungstechnik der Universität Bremen.

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. D. Popovic für die Betreuung und Förderung der Arbeit.

Herrn Prof. Dr.-Ing. W. Sobotta danke ich für die Übernahme des Korreferats.

Weiterhin danke ich Herrn Prof. Dr.-Ing. G. Schulz-Ekloff vom Institut für Angewandte und Physikalische Chemie für einige wertvolle Hinweise und die Bereitschaft zur gelegentlichen Expertenbefragung.

Letztlich danke ich allen meinen Kollegen am Institut, namentlich den Herren Dipl.-Ing. I. Hofmann und Dipl.-Ing. A.R. Tanha, für die jederzeit gezeigte Diskussions- und Hilfsbereitschaft.

Bremen, im August 1988
U. Meyer

Einleitung

1. Einleitung

Die folgenden Zeilen sind gewidmet den Eltern, die mich in die Welt gebracht haben und mich bis heute begleiten.

Ich bin dankbar für die Liebe, die ich von euch empfangen habe.

Es ist mir ein großes Glück, dass ich euch kennen darf.

Ich hoffe, dass ich euch bald wieder sehen darf.

Ich liebe euch sehr und werde euch immer dankbar sein.

Ich werde euch immer an euch denken.

Ich werde euch immer an euch denken.

Ich werde euch immer an euch denken.

Ich werde euch immer an euch denken.

Ich werde euch immer an euch denken.

Ich werde euch immer an euch denken.

Ich werde euch immer an euch denken.

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Ich

Meinen Eltern

gewidmet

Inhaltsverzeichnis

1.	Einleitung	1
2.	Grundlagen des Entwurfs und der Realisierung wissensbasierter Systeme	3
2.1	Methoden zur Repräsentation von Wissen	3
2.1.1	Objektorientierte Wissensrepräsentationsformen	4
2.1.1.1	Semantische Netzwerke	4
2.1.1.2	Objekt-Attribut-Wert-Tripel	7
2.1.1.3	Frames	9
2.1.2	Regelbasierte Wissensrepräsentation : Produktionsregeln	11
2.1.3	Logikbasierte Wissensrepräsentation	13
2.2	Werkzeuge zur Implementierung von Expertensystemen	17
2.2.1	Symbolverarbeitende Sprachen	19
2.2.1.1	LISP	19
2.2.1.2	PROLOG	21
2.2.2	Prozedurale Programmiersprachen	25
2.2.3	Einige Bemerkungen zu Rahmensystemen (Shells)	27
2.2.4	Allgemeine Leistungseinschätzung von Tools für Expertensysteme	31
2.3	Entwurfsschema für wissensbasierte Systeme	35
2.3.1	Der innere Ablauf der Realisierung : Rapid Prototyping	36
2.3.2	Der äußere Ablauf der Realisierung : Systementwurf auf der Basis einer erschöpfenden Problemanalyse	37
2.3.3	Rückkopplungen der Realisierung : Validation des Systems	39
3.	Probleme der Wissensakquisition und -umsetzung	41
3.1	Prinzipieller Ablauf der Knowledge-Engineering-Phase	42
3.2	Probleme der Wissensverarbeitung	45
3.2.1	Das Expertenproblem	45
3.2.2	Konsistenz und semantische Bedeutung einer Aussage	48

3.2.3	Randwertprobleme, Domänenbegrenzung und Verallgemeinerung spezieller Aussagen	50
3.3	Ansätze für die Einbeziehung von Modellen in die Wissensdarstellung und -verarbeitung	53
4.	Erkennung kritischer Prozeßzustände chemischer Reaktionen	57
4.1	Exemplarisch betrachtete Prozesse : Diskontinuierlich betriebene Rührkesselreaktoren	59
4.2	Die Anwendung konventionelle Methoden zur Lösung der Überwachungsaufgabe : Erkennung kritischer Zustände mit Methoden der Mustererkennung auf der Basis modellgestützter Meßverfahren	62
4.3	Erkennung kritischer Reaktionszustände durch Anwendung wissensbasierter Systeme	66
4.3.1	Motivation für den Einsatz von Expertensystemen	67
4.3.2	Struktur des wissensbasierten Prozeßüberwachungssystems	70
4.3.2.1	Problemstellung	71
4.3.2.2	Struktur des wissensbasierten Überwachungssystems auf der Grundlage eines heuristischen Ansatzes	72
4.3.2.3	Systemstruktur unter Integration eines einfachen Tendenzmodells	75
4.3.2.3.1	Auslegung des Tendenzmodells auf der Basis von Proportionalitäts-Zusammenhängen	77
4.3.2.3.2	Alarmklassifikation auf der Basis der Merkmalsextraktion mit dem Tendenzmodell	79
4.3.3	Akquirierung, Strukturierung und Formalisierung des zur Lösung der Überwachungsaufgabe benötigten Wissens	81
4.4	Realisierung von Demonstrationssystemen zur prinzipiellen Überprüfung des Systementwurfs	84
4.4.1	Konzipierung der Testsysteme	84
4.4.2	Ergebnisse der Simulationen	86
5.	Zusammenfassung und Ausblick	91
	Literatur	94
	Anhang	100

1. Einleitung

Eine wesentliche Aufgabe der Automatisierung technischer Prozesse stellt die Gewährleistung einer sicheren Führung der betrachteten Anlage dar. Treten Prozeßbedingungen auf, die eine ordnungsgemäße Durchführung des Prozesses oder, im Extremfall, die Sicherheit der Anlage gefährden, sind Überwachungssysteme gefordert, die eine so frühzeitige Erkennung kritischer Prozeßzustände gewährleisten, daß in einer verbleibenden Zeit Maßnahmen zur Sicherung eingeleitet und durchgeführt werden können.

Die Führung und Überwachung chemischer Reaktoren ist einer der wesentlichen Anwendungsbereiche für Methoden der Prozeßautomatisierung. Insbesondere für exotherme chemische Umsetzungen gilt dabei, daß extreme Reaktionsbedingungen, wie z.B. hohe Temperaturen und hoher Druck, die Prozeßführung charakterisieren. So können z.B. Fehlchargierungen oder zu hohe Einfülltemperaturen der Ausgangsstoffe eine Selbstbeschleunigung der Synthesereaktion oder das Anspringen unerwünschter Neben- oder Folgereaktionen auslösen, die eine erhebliche Erhöhung des Drucks im Reaktionsbehälter nach sich ziehen können. Die Folgen solchen Druckanstiegs können das Austreten von toxischen Stoffen in die Umgebung oder, im Extremfall, das Bersten des Reaktionsbehälters sein.

Für zahlreiche chemische Prozesse existieren keine, oder nur in der engen Umgebung eines Arbeitspunkts verifizierte, mathematischen Modelle. Ferner ist zu berücksichtigen, daß die meisten Prozeßvariablen nicht direkt meßbar sind, und daher nur die Information über einen Teilzustand der Anlage unmittelbar vorliegt. Konventionelle Überwachungsverfahren, die auf einer direkten Auswertung der Meßgrößen basieren [HUB82], unterliegen daher der Beschränkung, daß keine sichere Aussage über den Gesamtzustand der Anlage getroffen werden kann. Überwachungssysteme, die auf modellgestützten Meßverfahren basieren [SCH82], erzielen zwar durch die Berücksichtigung des zur Verfügung stehenden a-priori-Wissens über den Prozeß eine höhere Zuverlässigkeit, sind aber durch ihre Komplexität realisierungstechnisch schwierig umzusetzen. Mit der Anwendung diagnostischer Expertensysteme [BAT86] stehen für die Überwachungsaufgabe Methoden zur Verfügung, die eine direkte Auswertung des Erfahrungswissens des am Betrieb der Anlage beteiligten Personals zulassen. Diese wissensbasierten Methoden, die ihren

Ursprung im wesentlichen in der Behandlung medizinischer Diagnoseprobleme haben, erlangen auch für verfahrenstechnische Anwendungsprobleme eine zunehmende Bedeutung [LIB86].

Die vorliegende Arbeit behandelt das Problem der Erkennung kritischer Reaktionszustände chemischer Reaktoren mit Hilfe wissensbasierter Systeme. Im Gegensatz zu konventionellen algorithmengestützten Methoden wird dabei auf ein exaktes mathematisches Modell des Prozesses verzichtet, die in dieser Arbeit vorgeschlagenen Überwachungssysteme stützen sich allein auf das Erfahrungswissen über die betrachteten Reaktionen. Da die Wissensakquisition und -formalisierung die wesentliche Grundlage des Entwurfs von Expertensystemen bilden, stützen sich die Betrachtungen zum Anwendungsfall auf eine Diskussion der Wissenserwerbs- und umsetzungsproblematik aus praxisorientierter Sicht. Die entwickelten Prinzipien zur wissensbasierten Lösung der Überwachungsaufgabe werden durch die Realisierung einfacher Demonstrationssysteme anhand einfacher Beispiele überprüft. Eine Erweiterung der Ergebnisse auf kompliziertere Prozesse erscheint grundsätzlich möglich.

Im folgenden 2. Abschnitt werden zunächst die Grundlagen des Entwurfs und der Realisierung wissensbasierter Systeme dargestellt, wobei im ersten Teil die bekanntesten Methoden zur Repräsentation von Wissen, und im zweiten Teil die populärsten Programmierwerkzeuge zur Realisierung von Expertensystemen behandelt werden. Abschließend wird in Unterabschnitt 2.3 ein mögliches Entwurfsschema für wissensbasierte Systeme eingeführt. Im 3. Kapitel werden der prinzipielle Ablauf der Knowledge-Engineering-Phase dargestellt, sowie die wesentlichen Aspekte der Wissensverarbeitungsproblematik behandelt. Das 4. Kapitel stellt den zentralen Teil dieser Arbeit dar. Es werden zunächst die exemplarisch betrachteten Reaktionen beschrieben, wobei die mathematische Modellbildung der Bilanzgleichungen angegeben wird, auf die sich die algorithmischen Überwachungsverfahren stützen. Diese konventionellen Verfahren werden im folgenden, als Gegenüberstellung zur Anwendung von Expertensystemen, kurz zusammenfassend erläutert. Im Abschnitt 4.3 werden dann Entwurfsansätze zur wissensbasierten Lösung der Prozeßüberwachungsaufgabe vorgestellt, die anschließend anhand von Demonstrationssystemen prinzipiell überprüft werden.

Den Abschluß der Arbeit bildet eine Zusammenfassung der erzielten Ergebnisse, sowie ein Ausblick auf mögliche Erweiterungsmöglichkeiten der eingeführten Konzepte der Erkennung kritischer Prozeßzustände mit Hilfe wissensbasierter Methoden.

2. Grundlagen des Entwurfs und der Realisierung wissensbasierter Systeme

Die Entwicklung von Expertensystemen stützt sich auf den Ansatz, das Wissen über einen abgegrenzten Problembereich, die sogenannte Domäne, zu akquirieren, und in eine formalisierte Darstellungsweise zu überführen, die die programmiertechnische Realisierung zulässt [WAT85]. Im folgenden werden die bekanntesten Methoden zur Wissensrepräsentation und die am häufigsten verwendeten Werkzeuge zur Implementierung, sowie ein mögliches Entwurfsschema für wissensbasierte Systeme dargestellt.

2.1 Methoden zur Repräsentation von Wissen

Die Terminologie auf dem Gebiet der wissensbasierten Systeme kann gegenwärtig als noch nicht eindeutig bezeichnet werden, da in der Literatur immer wieder gleiche Begriffe für unterschiedliche Inhalte, sowie umgekehrt für gleiche Inhalte nicht übereinstimmende Begriffe verwendet werden. Das betrifft durchaus nicht nur Detailbezeichner, sondern ist auch in bezug auf übergeordnete Begriffe zu finden¹⁾.

Zumindest für die bekanntesten Methoden zur Repräsentation von Wissen läßt sich aber eine im wesentlichen einheitliche Nomenklatur voraussetzen. So unterscheidet man in der Klassifizierung von Wissensrepräsentationsformen die Kategorien objekt-, regel- und logikbasierte Darstellungsweise [HAR86]. Bezüglich der Nomenklatur, die innerhalb dieser Kategorien auftritt, läßt sich ferner feststellen, daß z.B. die Begriffswelt der Prädikatenlogik, als eine in sich geschlossene Theorie, unabhängig von der Terminologie der Expertensysteme festgelegt ist.

Die bekanntesten Wissensrepräsentationsmethoden, klassifiziert in die genannten Kategorien objekt-, regel- und logikbasierte Repräsentationsformen, sollen im folgenden beschrieben und ihre Merkmale charakterisiert werden.

1) Es wird z.B. der Begriff der Inferenzmaschine nicht für alle Klassen von Expertensystemen einheitlich verwendet.

2.1.1 Objektorientierte Wissensrepräsentationsformen

Objektorientierte Wissensrepräsentationsformen gründen sich auf die Absicht, den behandelten Problembereich abgeschlossen zu beschreiben, um mögliche Randwertprobleme auszuschließen. Das so erreichte Closed-World-Model ist dadurch in sich konsistent. Somit sind alle Lösungen bezüglich dieses Modells schlüssig, die essentielle Forderung für den Entwurf von Expertensystemen, ein streng abgegrenzter Arbeitsbereich, auch erfüllt, jedoch sind Lösungen bezüglich außerhalb der erfaßten Domäne befindlichen Objekten nicht existent.

Zur Verdeutlichung dieser Charakteristik ein Beispiel : Wird ein Expertensystem zur Diagnose von Fehlerquellen an einem Fahrrad auf der Basis von Zusammenhängen zwischen den Teilen des Fahrrads entworfen, so ist z.B. das Fehlen eines Objekts "Speichen" die Ursache dafür, daß ein möglicher Fehler "Speiche gebrochen" nicht analysiert werden kann; "Speichen" gehören nicht zur geschlossenen Welt "Fahrrad". Es ist kein Problem der Wissensrepräsentation, sondern vielmehr ein Problem der Wissensakquisition, die Zugehörigkeit des Objekts "Speichen" zur Domäne "Fahrrad" zu identifizieren.

Trotz dieses scheinbaren Nachteils gelten objektorientierte Repräsentationsformen in der Praxis als bewährt, insbesondere die hier im folgenden dargestellten Frames finden überwiegend Verwendung beim Aufbau von Wissensbasen über große, komplexe Problembereiche. Neben Frames gehören semantische Netzwerke und Objekt-Attribut-Wert-Tripel zu den bekanntesten objektorientierten Wissensrepräsentationsmethoden, die ebenfalls im folgenden behandelt werden sollen.

2.1.1.1 Semantische Netzwerke

Ein semantisches Netzwerk oder semantisches Netz (Bild 2.1) ist ein Repräsentationsschema, bestehend aus einer Sammlung von Knoten (nodes), die Objekte und Deskriptoren darstellen, und sie verbindenden Kanten (arcs, links), die die Relationen zwischen ihnen repräsentieren. Knoten und Kanten werden im allgemeinen benannt. Obwohl keinerlei Vorschriften für diese Benennung existieren, sind doch einige typische Konventionen dafür gebräuchlich geworden.

Objekte umfassen dabei sowohl physische Objekte, also Gegenstände, Menschen oder Tiere, als auch abstraktere Elemente, wie Handlungen, Ereignisse und Orte, oder beschreibende Faktoren, wie die Anzahl eines vorhandenen Objektes.

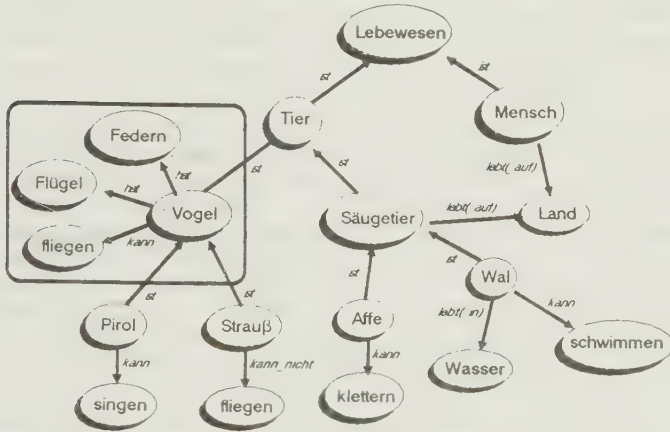


Bild 2.1 : Semantisches Netzwerk mit einem ausgewählten OAW-Tripel

Deskriptoren repräsentieren zusätzliche Informationen über Objekte, wie z.B. den Zustand und die Größe eines Objektes.

Die die Knoten verbindenden Kanten können jede Art von Relation zwischen Objekten, bzw. Objekten und Deskriptoren darstellen. Für diejenigen Relationen, die hierarchische Zusammenhänge charakterisieren, sind dabei die Bezeichnungen "ist" (is-a) und "hat" (has-a) gebräuchlich geworden. Ferner können Kantenbezeichner z.B. definierende Funktionen, oder Relationen, die heuristisches Wissen repräsentieren, sein.

Hierarchische Zusammenhänge seien hier in zwei Fälle unterschieden. Die "ist"-Relation wird verwendet, um die Verbindung zwischen dem Einzelfall, der Instanz, und seiner übergeordneten Klasse zu repräsentieren. Beispielsweise sagt die Relation "Pirol" "ist" "Vogel" aus, daß die Instanz "Pirol" der Klasse "Vogel" zugeordnet ist, "Vogel" "ist" "Tier" weist wiederum den Einzelfall "Vogel" der Klasse "Tier" zu. Die "hat"-Relation drückt dagegen die Verbindung zwischen Objekten und ihren Teilelementen oder Eigenschaften aus. So kann z.B. in der Beziehung

"Vogel" "hat" "Flügel" das Objekt "Flügel" nicht als Instantiierung der Klasse "Vogel" aufgefaßt werden, da ein einzelner Flügel kein Beispiel für einen Vogel ist. Eine "ist"-Relation über diese Objekte ergäbe somit keinen Sinn.

Kanten, die definierende Funktionen repräsentieren, stellen Relationen zwischen Objekten dar, die weder im Sinne des Klassenkonzeptes, noch als Eigenschaftsrelationen zu erfassen sind. Die Beziehung "Auto" "gebaut-in" "Bremen" ist ein Beispiel für eine definierende Relation, da Bremen weder die Instanz für ein Auto, noch Auto ein Teilelement von Bremen ist.

Heuristische Relationen sind in ähnlichem Sinne aufzufassen wie definierende Funktionen, im Gegensatz dazu repräsentieren sie jedoch Wissen, daß sich nicht auf Tatsachen, sondern mehr auf Erfahrungen stützt. Die dadurch entstehenden zusätzlichen Pfade im semantischen Netzwerk sind in erster Linie nützlich für die Vermeidung der Folgerung über viele Objekte, sowie die Behandlung von Ausnahmesituationen. So erleichtert z.B. die Relation "Regen" "verursacht" "Schirm-mitnehmen" den Zyklus deutlicher, als wenn über deterministische Relationen von einem Objekt zum anderen geschlossen werden muß.

Zu den Vorteilen des Repräsentationsschemas der semantischen Netzwerke gehört seine große Flexibilität, die es ermöglicht, nach Bedarf neue Knoten und Kanten zu ergänzen, bzw. bestehende Einträge zu modifizieren. Bezogen auf das Beispiel "Fahrrad", erfordert es keinen großen Aufwand, das fehlende Objekt "Speichen" dem diese Domäne beschreibenden Netzwerk über eine "ist-Teil-von"-Relation hinzuzufügen. Der Nachteil der Darstellung liegt in der "abstrakten Lesbarkeit" der Informationen, die repräsentiert werden, wobei mit zunehmender Komplexität der betrachteten Domäne dieses Problem wächst.

Ein weiteres Merkmal semantischer Netze ist, daß die Eigenschaften eines Knoten, der eine Klasse repräsentiert, auf einen diesem zugeordneten Einzelfall übertragen werden kann. Dieses Prinzip wird Vererbung (inheritance) genannt. So genügt es z.B. der Klasse "Vogel" durch eine "hat"-Relation die Eigenschaft "Federn" zuzuordnen, um über eine Instantiierung der Klasse mit dem Einzelfall "Pirol" über eine "ist"-Relation die Frage zu beantworten, ob ein Pirol Federn hat. Dieser Vererbungsmechanismus ist sehr effizient zu nutzen, da Redundanzen vermindert werden, jedoch tritt hierbei das Problem auf, daß die Behandlung von Ausnahmen erschwert wird. Liegt eine größere Anzahl von Ausnahmen vor, so müssen vielfältige und komplizierte Möglichkeiten zu ihrer Behandlung berücksichtigt werden. Ein einfaches Beispiel für die Einfügung einer Ausnahme in einem

semantischen Netz wäre es, das der Klasse "Säugetier" zugeordnete Attribut "lebt(-an)" "Land", für die Instanz "Wal" durch die Ergänzung der speziellen Eigenschaft dieses Einzelfalles zu komplettieren.

Das Wissensrepräsentationsschema der semantischen Netzwerke ist zurückzuführen auf den Versuch der Modellierung der Informationsspeicherung im menschlichen Gehirn. Die hier verwendete Begriffswelt stammt aus diesem Zweig der KI-Forschung.

2.1.1.2 Objekt-Attribut-Wert-Tripel

Ein Objekt-Attribut-Wert-Tripel ist aufzufassen als ein Ausschnitt aus einem semantischen Netz, der die Beziehungen zwischen einzelnen Objekten und ihren Eigenschaften beschreibt. Dabei treten einfache Relationen an die Stelle der vielfältigen Verbindungsglieder eines semantischen Netzes.

Objekte können wiederum sowohl physikalische als auch begriffliche Einheiten sein, Attribute sind die allgemeinen Charakteristika, die ein Objekt auszeichnen. Die Objekt-Attribut-Verbindung entspricht der "hat"-Relation, die Zuordnung Attribut-Wert der "ist"-Relation. Wesentliches Merkmal ist die Unterscheidung dieser beiden Glieder in den statischen, das unveränderliche Objekt mit seinen allgemeingültigen Attributen beschreibenden Teil, und der dynamischen Instanz, der Belegung des Attributs mit einem aktuellen Wert im Verlauf einer Systemkonsultation.

Objektbäume stellen die Anordnung und Beziehungen zwischen einzelnen Objekten dar, wobei das hierarchisch höchstwertige Objekt, die Wurzel, als Ausgangspunkt für Schlussfolgerungen und Informationsabfragen dient. Ein Objektbaum kann sowohl statisch in Form einer abstrakten, die allgemeingültigen Eigenschaften eines Objekts charakterisierenden Beschreibung das bearbeitende Wissen repräsentieren, als auch für die Instantiierung dieses Objekts mit einem bestimmten Wert die dynamische Abbildung der Zusammenhänge wiedergeben. Die Art der Relationen zwischen den Objekten im Baum ist nicht auf eine bestimmte Form festgelegt.

So kann z.B. ein Objektbaum für die Wurzel "Fahrrad" durch die Zuordnung der Einzelteile²⁾ aufgebaut werden (Bild 2.2). Möglich ist aber auch die Beschreibung, die die Ursachen von möglichen Defekten als Verbindungen verwendet (Bild 2.3).

Bild 2.2 : Statischer Objektbaum
in Form einer taxono-
mischen Zuordnung

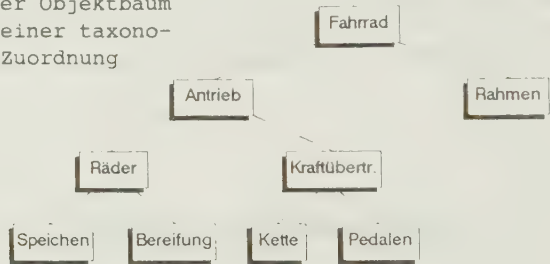
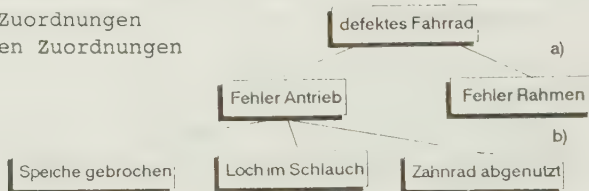


Bild 2.3 : Objektbaum einer Fehler-
hierarchie mit
a) Teile-Zuordnungen
b) kausalen Zuordnungen



Wissensbasierte Systeme, die durch einen Objektbaum repräsentiert sind, können, wie semantische Netzwerke, den Vorteil von Vererbungsmechanismen nutzen. Im Gegensatz dazu stehen solche Systeme, die formal nur um ein einziges Objekt herum aufgebaut sind. Man bezeichnet dieses Repräsentationsschema als Attribut-Wert-Paare. Mit Ausnahme der nicht realisierbaren Vererbungshierarchie verhält sich ein solches System aber wie ein OAW-Tripel-System.

Ein weiterer Aspekt der Wissensrepräsentation mittels OAW-Tripeln ist die Verfahrensweise beim Umgang mit vagem Wissen. So können OAW-Tripel bei der Instantiierung durch einen Konfidenzfaktor (certainty factor) ergänzt werden, der den Grad der Sicherheit einer Aussage repräsentiert. Ein solcher Konfidenz-

2) Im Fall einer derartigen hierarchischen Zergliederung wird auch der Begriff Taxonomie verwendet [MES85].

faktor stellt jedoch keinen Wahrscheinlichkeitsfaktor dar, sondern ist lediglich eine informelle Aussage darüber, inwieweit ein Zustand als tatsächlich richtig einzuordnen ist. In Mycin, einem der Urväter der Expertensysteme, werden diese Faktoren für die Einordnung einer Diagnose auf einer Skala von -1 (eindeutig falsch) bis +1 (eindeutig richtig) verwendet. Bei der Einbindung von Konfidenzfaktoren in einen Systementwurf ist jedoch zu beachten, daß der Experte, dessen Wissen abgebildet werden soll, sich der Schlüssigkeit seiner Aussage sicher ist, da eine falsch bewertete Aussage die Bedeutung des Wissens verfälschen kann. Hinsichtlich der Verarbeitung des Wissens durch verfeinerte Kontrollstrategien stellen Konfidenzfaktoren im Fall eindeutiger Zuordnungen jedoch einen wesentlichen Aspekt der Wissensgewinnung dar.

2.1.1.3 Frames

Eine komplexere objektorientierte Darstellungsform ist die der sogenannten Frames. Für den englischen Begriff hat sich noch kein passendes deutsches Äquivalent finden lassen, die direkte Übersetzung "Rahmen" erscheint nicht anspruchsvoll genug, um die Bedeutung richtig zu erfassen.

Ein Frame ist die Beschreibung eines Objekts, aufzufassen als eine Art "Eigenschaftsliste", die alle für das Objekt relevanten Informationen in sogenannten Slots (Abteilen) enthält (Bild 2.4). Die Einträge in den Slots können wiederum Werte von Attributen sein, wobei auch Vorbelegungen, d.h. Nominal- bzw. Defaultwerte, möglich sind. Weiterhin können die Abteile Verweise auf andere Frames enthalten, was die Berücksichtigung von Vererbungsmöglichkeiten zuläßt. Der wesentliche Unterschied zu semantischen Netzen und Objekt-Attribut-Wert-Tripeln besteht darin, daß Slots Regeln oder Prozeduren enthalten können, durch die man Werte erhält. Diese "Anweisungen" können Informationen anderer Slots oder auch anderer Frames miteinander kombinieren.

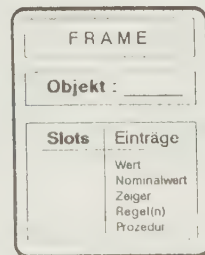


Bild 2.4 : Genereller Aufbau eines Frames

Frames ermöglichen durch die Kombination deklarativer und prozeduraler Wissensrepräsentation³⁾ eine umfangreichere Darstellung des Wissens über eine Domäne, sind jedoch schwerer zu entwickeln als semantische Netzwerke und OAW-Tripel. Die Komplexität von Frames kann die Verarbeitung von Wissen, in dem zahlreiche Ausnahmen vorkommen, abhängig von der Fragestellung an ein auf dieser Basis entwickeltes System, beeinflussen. So wird z.B. die Anfrage "Können Vögel fliegen ?" an ein Expertensystem, in dem Wissen gemäß Abbildung 2.5 enthalten ist, aufwendige Operationen erfordern, da der Nominalwert "ja" mit allen eventuellen Einträgen der Einzelfälle, die Ausnahmen repräsentieren, abgeglichen werden muß. Umgekehrt führt die Anfrage "Kann ein Pirol fliegen ?" über den Verweis zur Klasse "Vogel" direkt auf die Belegung mit diesem Defaultwert, da der Frame "Pirol" keinen Ausnahme-Eintrag enthalten wird.

FRAME	
Objekt : Vogel	
Slots	Einträge
Merkmal	Federn, Flügel : 2
fliegt	Standard : ja
Klasse	Tiere
Arten	Adler, Strauß, Pirol, ...
Große	(Regeln zur Ermittlung der Größe)
Lebensraum	(Verweis auf Tabelle für Einzelfälle)

Bild 2.5 : Beispiel für die Belegung eines Frames

Für die Repräsentation großer Domänen gelten Frames als bewährt, da damit eine kompakte und vollständige Darstellung erreicht werden kann. Werden Frames als Repräsentationsschema in einem wissensbasierten System zusammen mit einer anderen Wissensrepräsentationsform verwendet (z.B. ein Regelgebäude, das die objektbeschreibenden Frames umgibt), so spricht man von einem hybriden System. Ein Expertensystem, das allein auf der Wissensdarstellung durch Frames basiert, ist ein vollständig hybrides Wissenssystem.

3) Man spricht bei diesem Zusammenhang von dualer Semantik.

2.1.2 Regelbasierte Wissensrepräsentation : Produktionsregeln

Eine häufig verwendete Form, Wissen zu repräsentieren, ist die Darstellung in Produktionsregeln. Sie bestehen aus einem Bedingungsteil (Condition), der sogenannten Left-Hand-Side (LHS), und einem Folgerungsteil (Conclusion), der Right-Hand-Side (RHS) in der Form

```
IF < Condition(s) >
    THEN <.Conclusion(s) > .
```

Die Auswertung dieser Regeln wird in auf dieser Basis entwickelten Expertensystemen (Produktionsregelinterpreter) aufgrund eines sogenannten Work-Memory-Konzepts durchgeführt, wobei die Übereinstimmung der Prämisse einer Regel mit einem in der aktuellen Datenbasis (Work-Memory) vorliegenden Fakt die Ausführung des Konklusion der Regel, der Produktion, bewirkt. Das bekannteste Werkzeug, das auf diesem Ablaufmechanismus basiert, ist das in LISP geschriebene Werkzeugsystem OPS5, das zur Implementierung der DEC-Expertensysteme XCON, XSEL, XSITE, usw. benutzt wurde.

Produktionsregeln können sowohl für vorwärts- als auch für rückwärtsverkettete Inferenzmechanismen verwendet werden. Bei vorwärtsverketteten Systemen spricht man dann auch von Situation-Aktion-Regeln, rückwärtsverkettete Regeln repräsentieren eine Zustand-(erwartete-)Situation-Aussage. In Expertensystemen, die eine vermaschte Vorwärts-Rückwärts-Strategie zur Wissensverarbeitung abwickeln, ist es dadurch meistens erforderlich, für jede Inferenzrichtung eine äquivalente Regelmenge abzuleiten.

Regeln lassen sich als Abbildung der Beziehungen innerhalb einer Objekt-Attribut-Wert-Repräsentation auffassen, deren Verarbeitung dem folgenden Mechanismus genügt. Wird eine Regel durch eine Anfrage aktiviert, so werden alle OAW-Ausdrücke des Prämissenteils auf Richtigkeit überprüft. Trifft einer dieser Wenn-Tripel nicht zu, so wird die Abarbeitung der Regel beendet. Sind alle Bedingungen erfüllt, wird darauf geschlossen, daß auch der Dann-Teil richtig ist. Abbildung 2.6 zeigt eine Produktionsregel und ihre Transformation in ein OAW-Schema.

Ein weiteres Merkmal von Produktionsregeln ist, daß sie z.B. für mehrere Regeln, die ähnliche Inhalte repräsentieren, Variablen enthalten können. Damit ist es möglich, solche (konstanten) Regeln in einer "variablen" Regel zusammenzufassen. Ferner ist es möglich, Variablen dort zu verwenden, wo während einer Systemkonsultation dem Benutzer die Ermittlung von (Attribut-) Werten überlassen wird.

In Abbildung 2.7 ist die Substitution mehrerer Regeln durch die Einführung von Variablen anhand eines einfachen Beispiels dargestellt. Die "eliminierten" Informationen der zusammengefaßten Regeln müssen allerdings, z.B. in einer Tabelle, weiterhin verfügbar sein, um dem System die Anwendung der variablen Regel zu ermöglichen. Außerdem ist zu beachten, daß die Konklusionsteile der drei Regeln in der variablen Regel im Prämissenteil repräsentiert werden.

In Abbildung 2.6 ist die Folgeungsaussage des Beispiels mit einem Gewißheitsfaktor ("wahrscheinlich") versehen, der die Zuversicht in die Richtigkeit der Tatsache ausdrückt. Ebenso wie beim Repräsentationsschema der Objekt-Attribut-Wert-Tripel ist es möglich, in Produktionsregeln unsichere Relationen durch einen Konfidenzfaktor zu repräsentieren. In der dargestellten Regel ist z.B. nicht berücksichtigt, ob die Kabelzuleitungen in Ordnung sind. Die Vergabe von (Un-)Sicherheitsfaktoren erleichtert in Produktionsregelinterpretation somit den methodischen Umgang mit evtl. unvollständigem Wissen.

WENN die Batteriespannung normal ist UND der Lichtschalter auf "Ein" steht UND der Zustand der Glühbirnen normal ist,			
DANN ist es wahrscheinlich, daß das Fahrlicht brennt.			
	Objekt	Attribut	Wert
WENN	Batterie	Spannung	normal
	Schalter	Fkt. Zustand	"Ein"
	Glühbirne	Zustand	o.k.
DANN	Fahrlicht	Funktion	"An"

Bild 2.6 : Beispiel einer Produktionsregel und ihre äquivalente OAW-Tripel-Darstellung

R1	WENN das Gerät ein Staubsauger ist,
	DANN ist die Verwendung Teppiche reinigen.
R2	WENN das Gerät eine Waschmaschine ist,
	DANN ist die Verwendung Wasche reinigen
R3	WENN das Gerät eine Geschirrspülmaschine ist,
	DANN ist die Verwendung Geschirre reinigen.
variable Regel :	
	WENN das Gerät ein X ist UND
	die X Verwendung Y reinigen ist,
	DANN ist die Verwendung Y reinigen
Tabelle	
(Staubsauger) Verwendung (Teppiche)	
(Waschmaschine) Verwendung (Wasche)	
(Geschirrspülmaschine) Verwendung (Geschirre)	

Bild 2.7 : Beispiel für die Überführung mehrerer Regeln in eine variable Regel

Der wesentliche Vorteil der Wissensrepräsentation mittels Produktionsregeln ist, daß kombinatorische Explosionen vermieden werden und damit der Suchraum beschränkt bleibt. Ferner ist der Schlußfolgerungsmechanismus eines Produktionensystems determiniert, da durch Übereinstimmung von Bedingungs teilen "feuerbereite" Regeln erzeugt werden, deren Konklusionen den nächsten "Matching"-Zyklus initiieren usw., und somit eine Schlußfolgerungskette entsteht, die z.B. auch wieder zurückverfolgt werden kann. Als problematisch kann es sich erweisen, wenn Regeln eine relativ große Zahl von Bedingungen enthalten, da in einem so repräsentierten System unter Umständen allgemeingültige Beziehungen und Tatsachen nicht erfaßt werden können.

2.1.3 Logikbasierte Wissensrepräsentation

Auf die Darstellung von Wissen mit Hilfe des Prädikatenkalküls 1. Stufe stützt sich der Ablaufmechanismus der KI-Sprache PROLOG.

Die Prädikatenlogik ist eine Erweiterung der Aussagenlogik, in der Aussagen über Beziehungen zwischen Objekten, die die Grundelemente der Prädikatenlogik bilden, getroffen werden. Solche Aussagen, die die Werte "Wahr" oder "Falsch" annehmen können, werden Prädikate genannt. Prädikate können Aussagen über mehrere Objekte enthalten und zu verbundenen Ausdrücken verknüpft werden. Der Prädikatenkalkül (1. Stufe) besteht aus logischen Axiomen (Teilmengen der Formeln der Prädikatenlogik) und Inferenzregeln (Regeln zum Ableiten von Formeln). Grundregeln des Prädikatenkalküls sind z.B. modus ponens, modus tollens und Spezialisierung [APP85].

Die Prädikatenlogik basiert auf den jeweils abzählbaren Mengen von Grundsymbolen für logische Verknüpfungen, Variablen und Konstanten, Funktionen und eben Prädikaten. Die Verknüpfung dieser Elemente bezeichnet man als Formel. In Tabelle 2.1 sind die Grundsymbole und ihre Verknüpfungen zusammengefaßt. Der dort aufgeführte Existenz-Quantor wird auch Partikularisator, der All-Quantor Generalisator genannt.

Terme	
Konstanten	$a, b, c, \dots 1, 2, \dots$
Variablen	$x, y, z, x_1, \dots$ (in der Regel kleine Buchstaben vom Ende des Alphabets)
Funktionssymbole	$f(\cdot), g(\cdot), h(\cdot), \dots$
Prädikatsymbole	$P, Q, R, \dots$
atomische Formel, Atom	
$P, P(x), Q(a), \dots$	
Negation	
$\neg P, \neg Q(x), \dots$	
Disjunktion	
$P(x) \vee Q(x)$	
Konjunktion	
$P(x) \wedge Q(x)$	
Implikation	
$P(x) \supset Q(x)$	
Äquivalenz	
$P(x) \equiv Q(x)$	
Transformationen	
$P(x) \supset Q(x) \Leftrightarrow \neg P(x) \vee Q(x)$	
$P(x) \equiv Q(x) \Leftrightarrow P(x) \wedge Q(x) \vee \neg P(x) \wedge \neg Q(x)$	
$P(x) \equiv Q(x) \Leftrightarrow (P(x) \supset Q(x)) \wedge (Q(x) \supset P(x))$	
Quantoren	
Existenz-Quantor	$\exists x.P(x)$
All-Quantor	$\forall x.Q(x)$

Tabelle 2.1 : Syntaxvereinbarungen und Grunddefinitionen
der Prädikatenlogik

Ein Atom bzw. die Negation eines Atoms bezeichnet man als Literal. Eine Menge von disjunktiv verknüpften Literalen heißt Klausel, eine Menge konjunktiv verknüpfter Klauseln ist die Klauselmenge. Eine Klausel mit genau einem positiven Literal wird als Horn-Klausel bezeichnet.

Das erste theoretische Modell von PROLOG basiert auf der Tatsache, daß ein in dieser Sprache geschriebenes Programm durch (eine Menge von) Horn-Klauseln zu beschreiben ist. Es läßt sich

zeigen, daß prädikatenlogische Formeln sich in äquivalente Klauselmengen transformieren lassen [CL084]. Dazu sind die folgende Schritte notwendig.

Schritt 1 : Elimination von Implikationen

Es gilt (vergl. Tabelle 2.1)

$$P \supset Q = \neg P \vee Q ;$$

Schritt 2 : Auflösen von Negationen über verbundene Ausdrücke

Es gilt, analog zum DeMorgan'schen Gesetz in der Boole'schen Algebra, daß

$$\neg (P \wedge Q) = \neg P \vee \neg Q$$

und

$$\neg (P \vee Q) = \neg P \wedge \neg Q ,$$

ferner für Ausdrücke, die Quantoren enthalten, daß

$$\neg \exists x. P(x) = \forall x. \neg P(x)$$

und

$$\neg \forall x. P(x) = \exists x. \neg P(x) ;$$

Schritt 3 : Skolemisierung

Durch Einführung sogenannter Skolem-Konstanten und Skolem-Funktionen werden Existenz-Quantoren eliminiert :

a) Skolem-Konstante :

$$\exists x. P(x) \wedge Q(x, a) = P(k) \wedge Q(k, a) ;$$

b) Enthält die Formel einen All-Quantor, so muß eine Skolem-Funktion eingeführt werden :

$$\forall x. P(x) \supset \exists y. Q(x, y) = \forall x. P(x) \supset Q(x, f(x)) ;$$

Schritt 4 : Überführung in Pränex-Normalform

All-Quantoren werden an den Anfang der Formel gebracht und eliminiert, z.B. :

$$\begin{aligned}
 \text{a)} \quad & \forall x.P(x) \supset \forall y.Q(y) \supset R(x,y) = \\
 & = \forall x.\forall y.P(x) \supset Q(y) \supset R(x,y); \\
 \text{b)} \quad & (\forall x.P(x) \vee Q(x)) \wedge (\forall y.R(a,y) \vee S(y)) = \\
 & = (P(x) \vee Q(x)) \wedge (R(a,y) \vee S(y)) ;
 \end{aligned}$$

Schritt 5 : Überführung der quantoren-freien Formel in eine konjunktive Normalform

$$(P \wedge Q) \vee R = (P \vee R) \wedge (Q \vee R);$$

Schritt 6 : Überführung in Klauseln, z.B.

$$P(a) \wedge P(b) \wedge (P(x) \vee \neg Q(x,y) \vee \neg P(y))$$

wird transformiert in die aus drei Klauseln bestehende Klauselmenge

$P(a)$	erste Klausel,
$P(b)$	zweite Klausel,
$P(x) , \neg Q(x,y) , \neg P(y)$	dritte Klausel.

Bei dem zuletzt verwendeten Beispiel handelt es sich, wie leicht zu sehen, um eine Menge von Horn-Klauseln, da alle Klauseln genau ein positives Literal aufweisen. Für die Behandlung dieser Klauselmenge hinsichtlich der programmiertechnischen Umsetzung sei hier auf Abschnitt 2.2.1.2 verwiesen.

Die logikbasierte Wissensrepräsentation mittels des Prädikatenkalküls weist den Vorteil auf, daß die verwendeten Inferenzregeln (z.B. modus ponens) vollständig und korrekt sind. Allerdings muß berücksichtigt werden, daß eine Aussage über eine Tatsache nur die Werte "wahr" oder "falsch" annehmen kann, womit verbunden ist, daß Abfragen an das so repräsentierte Wissen z.B. nicht unmittelbar auf Eigenschaften eines Objekts gerichtet werden können. Die Suche nach Werten in einem logikbasierten System muß vielmehr immer darauf ausgerichtet sein,

den "Wahrheitsgehalt" einer Aussage zu überprüfen. Insbesondere im Fall des Auftretens vieler Ausnahmen erschwert diese Charakteristik die Verarbeitung des Wissens, da diese z.B. zu falschen Implikationen führen können. (Das klassische Beispiel ist hier die offensichtlich falsche Aussage "Jeder Vogel kann fliegen".) Tatsächlich läßt sich jedoch mit der Prädikatenlogik für geeignete Domänen eine hohe theoretische Eleganz bei der Formalisierung des Wissens erreichen.

2.2 Werkzeuge zur Implementierung von Expertensystemen

Zum Zeitpunkt der Realisierung der ersten bekannten Expertensysteme (z.B. DENDRAL, MYCIN) wurde der Entwicklungs-Software nicht sehr hohe Bedeutung beigemessen, da im Mittelpunkt der neuen Thematik die Philosophie des Problemlösungsansatzes stand. Leistungskriterien spielten seinerzeit eine eher untergeordnete Rolle. So wurden zunächst diejenigen Werkzeuge eingesetzt, die verfügbar waren. (was bis heute die "Nachwirkung" mitbringt, daß LISP nach wie vor für diesen Arbeitsbereich die populärste Programmiersprache ist.)

Das zunehmende Engagement auf vielfältigen Anwendungsfeldern führte dann zur Entwicklung neuer Systeme, zum einen auf dem Sektor "reiner" Programmiersprachen bzw. Programmierumgebungen, zum anderen wurde die Entwicklung von Shells, Expertensystemen mit "leerer" Wissensbasis, forciert, so ist z.B. MYCIN zum Rahmensystem EMYCIN (Empty MYCIN) weiterentwickelt worden [BUC84].

Die "Genese" von PROLOG brachte es ferner, gerade unter dem Aspekt kontinentaler Rivalitäten, mit sich, daß eine Art "Glaubenskrieg" zwischen den "LISP-Apologeten" in den USA auf der einen Seite und den pragmatischer denkenden Europäern ausbrach⁴⁾. In der Zwischenzeit haben sich die Ansichten jedoch etwas gewandelt, die zuvor verbreitete Meinung, Prolog sei ein "alter Flop in neuem Gewand", ist revidiert worden. Zahlreiche namhafte KI-Forscher in den USA sehen mittlerweile die europäischen Kollegen auf dem richtigen Weg [MCD80]. Eine Rolle spielte bei diesem Wandel allerdings mit Sicherheit auch die

4) Hintergrund dieser Gegensätzlichkeit ist die Tatsache, daß die, in den USA etabliertere, KI-Forschung in den Anfangszeiten dieser Disziplin auf Wissenschaften wie Psychologie, Sprachforschung, etc., aufbaute.

Tatsache, daß die Standardisierungsbemühungen für LISP stagnieren. Es existieren verschiedene LISP-Dialekte, die im wesentlichen an der jeweils verwendeten Hardware orientiert sind. Versuche, mit CommonLisp einen Standard zu erreichen, sollen in den USA auch deshalb mit Skepsis betrachtet werden, weil sich das Verteidigungsministerium dabei engagiert hat⁵⁾. In Europa steht derzeit eine französische Initiative (EuroLisp) zur Diskussion [PCS85].

Ein gewisser Pragmatismus (Verfügbarkeit von Werkzeugen und geeignetem, ausgebildetem Personal) ist sicherlich auch der Anlaß, daß in der "Frühzeit" Expertensysteme in "traditionellen" Programmiersprachen entwickelt wurden. So existieren Implementationen z.B. in FORTRAN oder ALGOL60. Neuere Tendenzen, insbesondere hinsichtlich des erweiterten Anwendungsrahmens von Expertensystemen (Realzeitfähigkeit), weisen allerdings darauf hin, daß wissensbasierte Systeme, trotz des höheren Programmieraufwands, zunehmend in prozeduralen Sprachen implementiert werden, um kritische Anforderungen, z.B. bezüglich der Rechenzeit, erfüllen zu können. (Parallel mit dieser Tendenz ist die Verbesserung der Leistungsfähigkeit expliziter KI-Werkzeuge zu beobachten, z.B. kompilierte PROLOG-Systeme oder spezielle Entwicklungsumgebungen für LISP.) Erleichtert wurde dieser Trend außerdem durch neue Programmierkonzepte, die die Entwicklung strukturierter Programmiersprachen nach sich zogen.

Im folgenden sollen die grundsätzlichen Konzepte einiger ausgewählter Werkzeuge kurz dargestellt werden, wobei aus dem Bereich der KI-Sprachen LISP und PROLOG berücksichtigt werden, und als Vertreter der prozeduralen Sprachen PASCAL, eine strukturierte Sprache, behandelt wird. Dabei gehen einige beispielhafte Bemerkungen zu FORTRAN, der ältesten bekannten Programmiersprache, in diesen Abschnitt ein. Auf die Rahmensysteme (Wissensrepräsentationssprachen und Shells) soll hier nur unter einigen allgemeinen Aspekten eingegangen werden.

5) Der aktuelle Stand ist, daß der Dialekt CommonLisp mittlerweile auch auf speziellen LISP-Maschinen, z.B. der Symbolics, implementiert ist.

2.2.1 Symbolverarbeitende Sprachen

Unter symbolverarbeitenden Sprachen versteht man diejenigen Programmierwerkzeuge, deren Entwurfsphilosophie sich auf die Absicht stützt, in erster Linie nicht numerische Größen, sondern symbolische Ausdrücke verarbeiten zu können. So ist z.B. LISP ursprünglich im Zusammenhang mit der Spieltheorie entwickelt worden (Schachprogramme). Es existieren mittlerweile einige Werkzeuge und Sprachmodifikationen, die auf der Symbolverarbeitung basieren, so z.B. die objektorientierten Sprachen SMALLTALK und POP-2. Da jedoch LISP und PROLOG die populärsten und daher meistverwendeten Werkzeuge dieser Kategorie sind, werden sie, exemplarisch für die Klasse der symbolverarbeitenden Sprachen, im folgenden dargestellt.

2.2.1.1 LISP

LISP (LISt Processing) [QUE84] ist eine in hohem Maße interaktive Sprache, die der Verarbeitung von symbolischen Ausdrücken dient. Dabei erfolgt die Erstellung und Bearbeitung eines Programms in dem Zyklus Lesen eines Ausdrucks, Auswertung des Ausdrucks und Ausgabe des Resultats (Read-Eval-Print-Schleife). Ausdrücke sind vornehmlich Listen, deren Elemente wiederum Listen oder Atome sind, bei deren Auswertung das erste Element der Liste als Operator interpretiert wird, der auf die restlichen Elemente der Liste (Operanden) angewendet wird. Daher spricht man im Zusammenhang mit LISP auch von applikativer Programmierung [BAR87]. Da man die Auswertung einer Liste durch den Operator QUOTE unterdrücken kann, ergibt sich als wesentliches Merkmal der Sprache, daß keine Unterscheidung zwischen Daten und Operationen gemacht wird. D.h. es ist erst bei der Betrachtung des ersten Listenelements möglich, festzustellen, ob eine Liste evaluiert werden soll oder nicht; Daten können als Programme interpretiert werden und vice versa. Weitere Merkmale von LISP sind die Methodik, Rekursionen als Grundkonzept der Beschreibung von Problemen und der Auswertung von Listen zu verwenden, sowie eine beliebige Tiefe und Breite der Verschachtelung von Listen zuzulassen.

Syntaktische Kontrollstrukturen basieren in LISP auf einigen wenigen Grundfunktionen, die zur Entwicklung komplexerer Funktionen verwendet werden. Die wichtigsten Grundfunktionen sind in Tabelle 2.2 zusammengefaßt, wobei Funktionsbezeichner und Syntaxbeschreibungen in bezug auf den Dialekt XLISP ausgewählt sind [BET84].

NAME	SYNOPSIS	BESCHREIBUNG	BEISPIEL, WERT
set, setq	(set <sym> '<expr>), (setq [<sym> <expr>]...)	Wertzuweisung an ein Symbol	(setq eins 1) 1
cons	(cons <expr1> <expr2>)	Konstruktion einer Liste	(cons 'a '(b c)) (a b c)
car	(car <expr>)	Erstes Element einer Liste	(car '(a b c)) a
cdr	(cdr <expr>)	Rest einer Liste	(cdr '(a b c)) (b c)
null	(null <expr>)	Ist eine Liste leer ?	(null '(a b c)) nil
atom	(atom <expr>)	Ist der Ausdruck ein Atom ?	(atom (cdr '(a b c))) nil
eq	(eq <expr1> <expr2>)	Sind zwei Ausdrücke identisch ?	(eq 'eins 'zwei) nil
eval	(eval <expr>)	Auswertung eines Ausdrucks	(eval '(summe 1 2)) 3
quote	(quote <expr>), '<expr>	Verhinderung der Evaluierung	(quote (a b c)) (a b c)
defun	(defun <sym><fargs><expr>...)	Definition einer Funktion	(defun summe (x y) (+ x y))
cond	(cond <pair>...), <pair> ::= <pred><expr>...	Bedingte Entwicklung von Paaren	(cond ((> x y)(summe x y) (t (setq y x))))
and, or	(and <expr>...), (or <expr>...)	Logisches UND und ODER	--
arithmetische Funktionen	(<op> <expr>...)	-	(+ 3 2) 5
relationale Funktionen	(<op> <expr1> <expr2>)	--	(= 4 5) nil
Ein-/Ausgabefunktionen	--	-	--

TABELLE 2.2 : EINIGE AUSGEWÄHLTE LISP-GRUNDFUNKTIONEN

In den meisten heute verfügbaren LISP-Systemen steht dem Anwender allerdings eine Vielzahl vordefinierter Funktionen und eine entsprechende Programmierumgebung zur Verfügung. So besteht z.B. der Dialekt InterLisp aus insgesamt ca. 10000 Funktionen. Eine andere Version, GCLisp, bietet z.B. eine Programmierumgebung, in der automatisch die notwendigen (schliessenden) Klammern gesetzt werden. Außerdem sei hier angemerkt, daß mittlerweile Hardware-Systeme existieren (z.B. Symbolics), die an die besonderen Erfordernisse dadurch angepaßt sind, daß spezielle LISP-orientierte Prozessoren (meist ausgeführt in Bit-Slice-Technik) verwendet werden und das Betriebssystem auf solchen Maschinen (meistens) ebenfalls in LISP geschrieben ist.

Der wesentliche Vorteil von LISP liegt in dem hohen Maß an Interaktivität, daß die Verwendung der Sprache bietet. Der entscheidende Nachteil ist die, zu keiner anderen Programmiersprache "passende", ureigene Datenstruktur, die die Anbindung von externen Modulen zumindest erheblich erschwert.

2.2.1.2 Prolog

Ursprünglich wurde PROLOG für Anwendungen im Bereich der Verarbeitung natürlicher Sprache als Erweiterung der ebenfalls von Colmerauer eingeführten Q-Systeme [COL70] entwickelt. Die Auslegung als Resolutions-Widerlegungs-Beweiser (Resolutionsprinzip, Robinson [ROB65]) führte jedoch dazu, zur Begrenzung des Suchraums Restriktionen einzuführen, die äquivalent zu Hornklauseln sind. Durch den auf dieser Basis erzielten Quasi-standard⁶⁾, gewann PROLOG zunehmende Bedeutung für Anwendungen im Bereich der Künstlichen Intelligenz und der Expertensysteme, speziell als "Rapid-Prototyping"-Werkzeug [SCN86].

Die Abarbeitung von Hornklauseln erfolgt in PROLOG nach den Mechanismen Unifikation, Matching und Backtracking [EST87].

6) Man spricht hier von der Edinburgh-Syntax.

Dabei basiert die Verarbeitung des repräsentierten Wissens auf drei Grundprinzipien :

1. Der Darstellung von (mit Sicherheit) wahren Aussagen durch Fakten;
2. Der Darstellung von Ableitungsmechanismen für implizit vorhandenes Wissen durch Regeln;
3. Zielanfragen, d.h. zu beweisenden Aussagen, an das durch Fakten und Regeln im Programm repräsentierte Wissen.

PROLOG ist eine deklarative Programmiersprache, daher sind Datenstrukturen, wie sie in imperativen Sprachen verfügbar sein müssen, nicht erforderlich. Einzige Datenstruktur ist hier vielmehr der PROLOG-Term, der entweder eine Konstante (in der Regel eine Folge von Kleinbuchstaben = Atom), ein (quotierter) String, ein Zahlenwert, eine Variable (erstes Zeichen Großbuchstabe), eine Struktur, gebildet aus Funktor und Argumenten, oder eine Liste ist, wobei die Argumente einer Struktur bzw. die Elemente einer Liste selbst wieder Terme sind. Dabei wird zwischen Funktoren gleicher Bezeichnung mit unterschiedlicher Stelligkeit (Arity), d.h. ungleicher Zahl von Argumenten, differenziert. Funktoren - in Anlehnung an die Prädikatenlogik auch Prädikatssymbole genannt - ohne Argumente (nullstellige Prädikate) sind Konstanten, Terme ohne Variablen Grundterme.

Um die Abarbeitung von Klauseln darzustellen, sei hier nochmals das Beispiel aus Abschnitt 2.1.3 (Schritt 6) betrachtet. Die dort hergeleitete Klauselmenge, bestehend aus drei Einzelklauseln, lautete

$$\begin{aligned} &P(a) \\ &P(b) \\ &P(x) , \neg Q(x,y) , \neg P(y). \end{aligned}$$

Zu beachten ist, daß die Kommata in der dritten Klausel hier lediglich Trennzeichen für die Literale der Klausel sind, der gesamte Ausdruck ist eine disjunktive Verknüpfung. Da alle drei Klauseln genau ein positives Literal enthalten, handelt es sich bei dem Beispiel um Hornklauseln. Die Syntax-Notation in PROLOG genügt nunmehr der Vorschrift, die nicht negierten Literale einer jeden Klausel links eines Trennzeichens, das hier die "Umkehrung" einer Implikation bezeichnet, sowie die negierten Literale ohne die Negation konjunktiv verknüpft rechts des Trenn-

zeichens zusammenzufassen, ferner jede Klausel durch einen Punkt abzuschließen. Damit lautet obiges Beispiel in PROLOG-Schreibweise

```
p(a) :- .
p(b) :- .
p(X) :- q(X,Y),p(Y).
```

Die Literale links des Trennzeichens werden Kopf (Head) der Klausel, alle Literale rechts davon Rumpf (Body) genannt. Da alle Klauseln das gleiche (einstellige) Prädikat im Head haben, bilden diese drei Klauseln zusammen eine Prozedur. Bei den ersten beiden Klauseln handelt es sich um Fakten, da der Body den Wert "wahr" hat. Üblicherweise wird in der PROLOG-Syntax dafür

```
p(a).
p(b).
```

geschrieben. Die dritte Klausel stellt eine Regel dar, die semantisch wie folgt interpretiert werden kann :

für alle X und Y gilt :

$p(X)$ ist wahr, wenn $q(X,Y)$ und $p(Y)$ wahr sind⁷⁾

(deklarative Interpretation).

Eine Zielfrage (Goal) stößt den Abarbeitungsmechanismus von PROLOG an⁸⁾. Dabei wird das Goal mit dem Head der ersten Klausel einer Prozedur mit dem gleichen Funktor (mit gleicher Arity) gematcht, d.h. die "Muster" miteinander verglichen. Werden bei dieser Unifikation Variablen als Argumente verwendet, spricht man von einer Unifikationssubstitution. Für den auf diese Weise gematchten Head wird nun durch Bestätigung des Body ein Beweis der "Wahrheit" der Zielfrage versucht, wobei nacheinander jedes Teilziel⁹⁾, unter Übernahme der Variablensubstitution vom Head (sofern erfolgt), abgearbeitet wird (Depth-First-Search). Kann dabei ein Teilziel nicht bestätigt werden, so wird versucht, das vorherige Subgoal (mit einer alternativen Lösung) erneut zu matchen, sofern keine weiteren Klauseln innerhalb dieser Prozedur existieren, ansonsten wird die Unifika-

7) Das bekannteste in der Literatur für diese Klauselmengen angeführte Beispiel ist : Jeder (X) ist ein Mensch (p), wenn er eine Mutter (q) hat, die ein Mensch (Y) ist, und Adam (a) und Eva (b) sind Menschen [CLO84].

8) Die Verwaltung von Zielfragen wird über Stacks abgewickelt.

9) Die einzelnen Prädikate des Body einer Klausel.

tion der nächsten Klausel einer Prozedur versucht. Dieser Vorgang wird Backtracking genannt, man unterscheidet dabei gemäß den beiden oben genannten Fällen zwischen Deep und Shallow Backtracking. Der ganze Ablaufmechanismus wiederholt sich, bis alle Teilziele für ein Goal erfüllt sind oder keine alternativen Lösungen mehr für Teilziele aufgestellt werden können (fail).

Das Backtracking kann dabei auf zweierlei Arten beeinflusst werden. Zum einen ist es möglich, durch Einfügung eines Teilziels CUT ("!") die Abarbeitung abzubrechen, zum anderen können durch künstliches Fehlersetzen eines Goals alle möglichen Lösungen berechnet werden. Gerade diese Nichteindeutigkeit unterscheidet PROLOG von herkömmlichen imperativen Programmiersprachen.

Abschließend hier noch die Betrachtung des Abarbeitungsmechanismus am obigen Beispielpogramm.
Bestätigt werden soll das Goal

?- p(d).¹⁰⁾

In den ersten beiden Schritten wird versucht, die Zielanfrage mit p(a) und p(b) zu matchen. Da (die Konstante) "d" nicht mit den Konstanten "a" und "b" unifiziert werden kann, führt Shallow Backtracking zur dritten Klausel, was die Unifikationssubstitution $X = d$ liefert. Damit ist nun das Unterziel $q(d,Y)$ zu erfüllen¹¹⁾, was z.B. mit der zweiten Unifikationssubstitution $Y = b$ verbunden ist. Das so instantiierte zweite Teilziel p(b) wird zunächst mit der ersten Klausel, sodann wiederum über Shallow Backtracking mit der zweiten Klausel gematcht, womit die Zielanfrage durch Erfüllung aller Subgoals bestätigt wird. Daraus ist zu erkennen, daß die Fakten der Prozedur das Abbruchkriterium für die in der Regel enthaltene Rekursion darstellen.

Der wesentliche Vorteil der Programmiersprache PROLOG ist darin zu sehen, daß der Ablaufmechanismus - wahrscheinlich bei der Entwicklung der Sprache noch unberücksichtigt - Kontrollstrategien, wie sie in bestimmten Expertensystemapplikationen erforderlich sind, unterstützt. So sind z.B. Probleme, die durch eine reine Rückwärtsverkettung (zielgetriebene Suche) zu lösen sind (z.B. Planungsaufgaben), durch den PROLOG-Ablaufmechanismus direkt nachvollziehbar.

10) "?-" ist in den meisten PROLOG-Systemen der Goal-Prompt.

11) Vorausgesetzt wird, daß $q(X,Y)$ ebenfalls durch eine Prozedur berechnet wird.

Ferner ist als Vorteil anzuführen, daß die Sprache, z.B. gegenüber LISP, sich durch eine weitaus höhere Selbstdokumentierbarkeit auszeichnet¹²⁾, was mit Sicherheit auch auf den historischen Ursprung, die Q-Systeme, zurückzuführen ist.

Der Nachteil von PROLOG ist darin zu sehen, daß dem Anwender nur geringe Möglichkeiten gegeben sind, die Ablaufsteuerung eines Programms zu beeinflussen. Damit kann sich, bei ungeeigneter Problemstellung, eine erschöpfende Suche im Programmablauf ergeben, die, gerade bezüglich zeitkritischer Probleme, aufgrund der resultierenden kombinatorischen Explosion die Vorteile der Sprache reduziert.

2.2.2 Prozedurale Programmiersprachen

Der Einsatz prozeduraler Programmiersprachen in der Expertensystem-Entwicklung wird relevant, wenn realzeit-orientierte Aufgaben zu lösen sind, da die zuvor beschriebenen, an der KI-Problematik orientierten Werkzeuge ungeeignet sind, wenn das (Hardware-)Zielsystem mikroprozessorbasiert aufgebaut bzw. periphere Hardware mit dem Rechner verbunden werden soll¹³⁾.

Als Beispiel sei hier das medizinische Diagnosesystem PUFF [AIK84] genannt, das - ursprünglich in LISP implementiert - in BASIC umprogrammiert wurde, um die Kopplung mit einer "Meß-Hardware" zu gewährleisten.

Höhere Programmiersprachen stellen für derartige Aufgaben ein adäquates Entwicklungs-Werkzeug dar, da - im Gegensatz zu maschinenorientierter Programmierung (Assembler) - ein von der verwendeten Rechner-Hardware unabhängiger Standard exi-

12) Es liegt natürlich am Programmierer, die "Namensgebung" in einem Programm so lesbar wie möglich zu gestalten. Ferner ist diese Eigenschaft an die Entwicklungsumgebung des verwendeten Dialekts gebunden.

13) Das gilt in besonderem Maße für ältere Dialekte von LISP und PROLOG, die hinsichtlich arithmetischer Operationen Schwächen aufweisen, und nicht die Möglichkeit bieten, periphere Elemente anzusprechen.

stiert¹⁴⁾. Dieser Vorteil auch gegenüber KI-Werkzeugen, insbesondere im Vergleich zu LISP mit seinen zahlreichen Dialekten, erleichtert die Dokumentation der Software und die spätere Wartung der Programme in der Produktion.

Als ein Beispiel sei hier die älteste bekannte problemorientierte Programmiersprache FORTRAN [MAC83] genannt, welche 1954 - 57 von IBM entwickelt wurde. FORTRAN steht für FORmula TRANslation und dient der Verarbeitung mathematischer Ausdrücke. Aufgrund ihres hohen Alters ist die Sprache an hardware-bedingte Restriktionen gebunden, die insbesondere die Selbstdokumentierungsfähigkeit erschweren. So hat sich z.B. der Programmierer bei der Wahl von Variablennamen an begrenzte Zeichenkettenlängen zu halten. Ferner erschwert das Fehlen von (explizit vorhandenen) "logischen Schleifen" im Sprachumfang die Lesbarkeit eines FORTRAN-Programms¹⁵⁾. Ein weiteres - ebenfalls historisch bedingtes - Merkmal ist die Orientierung der Ein-/Ausgabe am "Lochkartenformat", die die interaktive Nutzbarkeit einschränkt, allerdings die Dateiverwaltung erleichtert. Es ist allerdings anzumerken, daß die Standards von Zeit zu Zeit modifiziert werden, um den veränderten Bedingungen (z.B. leistungsfähigere Hardware) Rechnung zu tragen.

Die Mächtigkeit von FORTRAN liegt in der numerischen Stärke der Sprache, so ist hier die Möglichkeit gegeben, Rechenoperationen mit erhöhter Genauigkeit (z.B. doppelter Wortlänge) durchzuführen, was den Quantisierungsfehler begrenzt. Diese Stärke ist dadurch bedingt, daß FORTRAN für den industriellen Einsatz auf Großcomputern entwickelt wurde. Durch den - im Gegensatz zu moderneren prozeduralen Sprachen - geringen Sprachumfang ist FORTRAN einem großen Kreis von Anwendern zugänglich geworden und in hohem Maße portabel. Letztlich erweist es sich als Vorteil, daß FORTRAN durch entsprechende Erweiterungen realzeitfähig ist.

Die Entwicklung neuer Entwurfskonzepte und leistungsfähiger Kleinrechner zog es nach sich, daß Sprachen entwickelt wurden, die modulare und strukturierte Programmierung ermöglichen. Ein Ergebnis ist die Programmiersprache PASCAL [RIE85], die von N.Wirth Anfang der 70er Jahre in Zürich entwickelt wurde und ursprünglich als reine Lernsprache konzipiert war. Aufgrund dieser Philosophie fehlt in PASCAL z.B. die Möglichkeit der

14) Vom jeweiligen Hersteller abhängig, differiert der Sprachumfang einer Hochsprache; dabei handelt es sich allerdings um Erweiterungen des genormten Standards.

15) FORTRAN wird von manchen deshalb als "GOTO-Sprache" bezeichnet.

doppelten Rechengenauigkeit. PASCAL ist als Abbildung der Nassi-Shneiderman-Diagramme, auch Struktogramme genannt, aufzufassen, daher gilt die Verwendung des GOTO-Statements, obwohl im Sprachumfang vorhanden, als "verboten"¹⁶⁾.

PASCAL ist hochgradig flexibel und bietet dem Programmierer vielfältige Möglichkeiten in der Anwendung. So können benutzerdefinierte Datenstrukturen, Variablentypen und Indexmengen von Feldern deklariert, sowie verschiedene Typen von logischen Schleifen verwendet werden. Wesentlicher Vorteil der Sprache ist die Möglichkeit der dynamischen Speicherbelegung zur Laufzeit mittels sogenannter Pointer, so ist z.B. der PROLOG charakterisierende Ablaufmechanismus mit diesen Zeigervariablen nachzubilden.

PASCAL ist aufgrund seiner einfachen Struktur und der Möglichkeit des Einsatzes auf Kleinrechnern weit verbreitet und wird mittlerweile auch im industriellen Bereich eingesetzt.

Allerdings weist PASCAL auch einige Nachteile bzw. Fehlermöglichkeiten auf. Die bereits anfangs erwähnte Schwäche in der Numerik, sowie die in der Eingabe von Daten und bei der Dateibehandlung auftretenden Probleme, begrenzen die Verwendungsfähigkeit der Sprache. Die Portabilität von PASCAL-Programmen ist eingeschränkt durch diverse Spracherweiterungen bei Compilern verschiedener Hersteller, jedoch sind am Standard orientierte Entwicklungen durchaus übertragbar.

2.2.3 Einige Bemerkungen zu Rahmensystemen (Shells)

Expertensysteme mit "leerer" Wissensbasis sind in großer Zahl auf dem Markt verfügbar und stellen bis zu einem gewissen Grad eine mögliche Alternative zur vollständigen Implementation eines wissensbasierten Systems dar. Die Angebotspalette ist nicht nur groß, sondern auch qualitativ zu klassifizieren. So existieren zum einen kleinere Systeme, die im allgemeinen bis zu 100 Regeln verarbeiten können, zum anderen sind große hybride Systeme verfügbar, die entsprechend teuer sind und deshalb eine sorgfältige Nutzenanalyse durch den Anwender erfordern.

16) FORTRAN ist an der Flußdiagramm-Technik orientiert.

Kleinere Rahmensysteme [KEL86], auch als Shells bezeichnet, sind in der Regel so aufgebaut, daß eine bestimmte Wissensrepräsentationsform für den Aufbau der Wissensbasis verwendet werden muß. In den meisten Fällen sind dies Produktionsregeln, da die Formulierung des Wissens über eine Domäne in Wenn-Dann-Aussagen einer natürlichsprachigen Beschreibung am nächsten kommt. Der realisierte Schlußfolgerungsmechanismus ist im allgemeinen einfach aufgebaut, der Benutzer hat wenig oder keinen Einfluß auf die Konfliktlösungsstrategie¹⁷⁾.

Shells zeichnen sich dadurch aus, daß auch Nicht-Programmierer in relativ kurzer Zeit den Umgang mit ihnen erlernen können. Ferner ist die Entwicklungszeit für ein Expertensystem mit einem solchen Werkzeug natürlich geringer, da lediglich die Wissensbasis implementiert werden muß. Somit eignen sich Shells speziell für schnelle Prototypentwürfe und die Einarbeitung in die betrachtete Domäne.

Die mangelnde Flexibilität einer Shell läßt den Entwickler in der Phase der Verfeinerung des repräsentierten Wissens und des daraus realisierten Systems aber schnell die Grenzen des Werkzeugs erkennen. Daher ist die Neuprogrammierung des Expertensystems häufig erforderlich, zumal das Verhältnis des zeitlichen Aufwandes zwischen Wissensakquisition und Systemimplementation - der Erwerb des Wissens ist die zeitaufwendigste Aufgabe der Entwicklung - eine zeitintensivere, aber dafür ausgereifere Realisierung rechtfertigt.

Größere Rahmensysteme, z.B. BABYLON [DIP85] oder TWAICE [KOR85], [MES85], sind dadurch charakterisiert, daß sie keine Werkzeuge für den Programmierer, sondern für den Wissensingenieur sind. Sie bieten die Möglichkeit, verschiedene Wissensrepräsentationsformen alternativ bzw. komplementär zu verwenden. Aufgrund dieser Hybridität bezeichnet man große Rahmensysteme auch als Wissensrepräsentationssprachen. Der Zweck der Kombination verschiedener Repräsentationsmechanismen liegt in der Absicht, unterschiedliche Formen von Wissen in der jeweils bestgeeigneten Art abzubilden. So läßt sich z.B. das Erfahrungswissen eines Experten am natürlichsten in Produktionsregeln darstellen (Wenn vorliegende Situation Dann geeignete Aktion), während Zustands- bzw. gegenständliche Beschreibungen über Objekte eher in Form von Frames am geeignetsten repräsentiert werden, und Zusammenhänge innerhalb der Domäne, die definitorisch zu erfassen sind, durch die Prädikatenlogik am

17) Meist hat der Benutzer lediglich die Möglichkeit, zwischen Vorwärts- und Rückwärtsverkettung zu wählen, wobei eine lexikalische Konfliktlösung realisiert ist, d.h. die erste "feuerbereite" Regel wird ausgeführt.

besten zu beschreiben sind [DIP85]. Neben den sich ergänzenden Repräsentationsformalismen stellen Wissensrepräsentations-sprachen dem Anwender in der Regel verschiedene Inferenzmechanismen zur Verfügung.

Große Rahmensysteme zeichnen sich dadurch aus, daß dem Benutzer eine transparente Oberfläche zur Verfügung steht, die den Umgang mit dem Werkzeug erleichtert und sich durch gute Erklärungsfähigkeiten der bearbeiteten Prozesse auszeichnet. Damit kann erreicht werden, auch große Wissenbasen übersichtlich zu gestalten und den Aufwand bezüglich Aufbau und Pflege zu reduzieren. Außerdem weisen sie einen höheren Grad der Offenheit auf als kleinere Shells, sind jedoch durch ihre Paradigmen-orientiertheit¹⁸⁾ an Restriktionen gebunden, die nur durch den Einsatz reiner Programmiersprachen zu umgehen sind. Ferner erweist es sich in den meisten Fällen als Nachteil, daß nicht nur die Kosten für die Software selbst, wie oben bereits erwähnt, relativ hoch sind, sondern außerdem große Rahmensysteme nur auf Spezialmaschinen implementiert sind, die ebenfalls im Bereich des Vielfachen der Kosten z.B. eines PC liegen. So werden Expertensysteme, die mittels einer Wissensrepräsentationssprache entwickelt werden, in dem Fall, daß "Experten-Kosten" eingespart werden sollen, lediglich eine Verlagerung der Kosten bewirken, im extremsten Fall durch das ungeeignete Zielsystem nicht in der Produktion realisierbar sein.

Als Zwischenstufe, innerhalb des Gegensatzes reiner Programmiersprachen contra Rahmensysteme, sind sogenannte schmal-spurige Werkzeuge [HAR86] zu betrachten.

Ein Beispiel für diese Klasse von Systemen ist OPS5 [BRW85], in der Originalversion in LISP implementiert, welches als spezielles Werkzeug für (vorwärtsverkettete) Produktionsregelsysteme zu kategorisieren ist. Diese Klassifizierung ist nicht unbedingt eindeutig, jedoch wird die Sprache hauptsächlich für die Entwicklung von vorwärtsverketteten, regelbasierten Systemen verwendet. Musterbeispiel ist das Konfigurations-Expertensystem XCON und seine "Ableger" [MER86].

Ursprüngliche Absicht seiner Entwickler (Carnegie Mellon Univ.) war es, die Sprache als Werkzeug für die Erforschung des menschlichen kognitiven Verhaltens im Bereich der Psychologie

18) Eine Forderung, die an Werkzeugsysteme gestellt werden muß, um sie für konkrete Aufgaben einsetzen zu können, ist die Orientierung des Rahmensystems an einem der Beratungsparadigmen.

einzusetzen. Hintergrund dieses Zwecks ist die Hypothese [NEW72], daß Produktionsregeln das menschliche Erkennungsvermögen hinreichend genau beschreiben.

Die Wissensdarstellung erfolgt in OPS5 mittels Regeln, den sogenannten Produktionen, die als Wenn-Dann-Aussagen, transformiert in ein OAW-Schema (vgl. Abschn. 2.1.2, Abb. 2.6), aufzufassen sind. Fakten sind aktuelle Einträge in den Arbeitsspeicher (Work-Memory), die als Objekte mit Attributen und Werten repräsentiert werden. Der Schlußfolgerungsmechanismus (Inferenz) läuft nun derart ab, daß zu Beginn einer Konsultation - der Arbeitsspeicher ist leer - Fakten in das Work-Memory eingetragen werden¹⁹⁾. Nach diesen Eintragungen werden die Regeln, bzw. Gruppen von Regeln, mit dem Arbeitsspeicher verglichen. Diejenigen Regeln, deren Wenn-Teile mit den eingetragenen Fakten übereinstimmen²⁰⁾, bilden sodann die sogenannte Konfliktmenge. Aus der Menge der zutreffenden ("feuerbereiten") Regeln wird aufgrund einer ausgewählten Strategie die aktionsausführende Regel ausgewählt.

Derartige Schmalspur-Werkzeuge wie OPS5 haben sich in streng abgegrenzten Anwendungsbereichen durchaus bewährt (siehe Beispiel DEC-Expertensysteme), jedoch ist die mangelnde Flexibilität auch dieser Systeme der Grund für keinen verbreiteten Einsatz.

Zusammenfassend läßt sich feststellen, daß Rahmensysteme den Entwicklungsaufwand der Expertensystemerstellung sowohl bei der Verwendung kleiner Shells für Demonstrationssysteme, als auch bei der Nutzung von Wissensrepräsentationssprachen für größere Projekte reduzieren. Es ist beim Übergang vom Prototyp auf ein Produktionssystem im Einzelfall aber zu prüfen, ob die - durch die am Anwendungs-Paradigma orientierten - Grenzen des Werkzeugs nicht eine Reimplementation in einer reinen Programmiersprache (oder mittels eines anderen Werkzeugsystems) erforderlich machen.

19) gebräuchlich ist ein Fakt "(start)".

20) man spricht dabei von feuerbereiten Regeln.

2.2.4 Allgemeine Leistungseinschätzung von Tools für Expertensysteme

Bei der Beurteilung der Leistungsfähigkeit der behandelten Programmierwerkzeuge für Expertensysteme stellt sich zunächst die Frage, ob eine Untersuchung überhaupt ein objektiv sinnvolles Vergleichsergebnis liefern kann. Naheliegend ist die Einschätzung, daß prozedurale Sprachen nicht mit KI-orientierten Sprachen bezüglich der gleichen Kriterien beurteilt werden können, da bereits die Entwurfsphilosophien der Sprachentwicklung nicht vergleichbar sind. Jedoch selbst scheinbar vergleichbare Werkzeuge wie LISP und PROLOG weisen so unterschiedliche Stärken auf, daß eine komparative Analyse ohne Berücksichtigung des behandelten Problems fragwürdig erscheint.

Benchmark-Tests, für prozedurale Sprachen durchaus üblich, wenn es um den Vergleich von Rechenzeiten geht, sind für KI-Werkzeuge nicht unbedingt gebräuchlich, wenn auch vereinzelt durchgeführt²¹⁾. Das Kriterium der "Logic Inferences per Second" (LIPS) ist ebenfalls nur als qualitatives Beurteilungsmerkmal zu charakterisieren, da die unterschiedlichen Ablaufmechanismen und die unvorhersehbare Suchtiefe bei Implementationen mit verschiedenen Werkzeugen keine eindeutige Aussage zulassen. Laufzeiten sind letztlich daran gebunden, was mit einer Sprache erreicht werden soll. So ist z.B. die hohe Interaktivität von LISP, die beliebige Änderungen des Programms zur Laufzeit zuläßt, mit keiner anderen Sprache vergleichbar²²⁾.

Der Programmieraufwand bezüglich der "Perfektion" eines Programmiers im Umgang mit einer Sprache ist folgerichtig ebenfalls nur subjektiv zu beurteilen, da die Abschätzung der Entwicklungszeit eben auch an die Person dieses Programmiers gebunden ist. Das gilt in gleicher Weise für den Aufwand der "Vorbereitung", d.h. für den Zeitraum der nötig ist, um

21) Ein Benchmark-Test für verschiedene PROLOG-Versionen, LISP und POP-2 (Dialekte jeweils unbekannt) ist in [WAR77] dokumentiert, ist aber eher als Werbung für das eigene Produkt aufzufassen.

22) Programmieraufgaben, die off-line zu lösen sind, sollten nach Möglichkeit an keine bestimmte Sprache gebunden sein, um sämtliche Vorteile des besten Werkzeugs auszunutzen.

ausreichende Kenntnisse im Umgang mit einem Werkzeug zu erlernen. Ein eingeschworener FORTRAN-Programmierer wird sich, um z.B. PROLOG zu erlernen, von der FORTRAN-Philosophie lösen müssen, um das andersartige Werkzeug PROLOG optimal für das behandelte Problem einsetzen zu können.

Das dritte übliche Beurteilungskriterium von Programmiersprachen ist die Frage des Speicherplatzbedarfs. Durch die Entwicklung im Hardware-Bereich tritt dieses (eventuelle) Problem allerdings in den Hintergrund, da z.B. auf Personal Computern auch (fast) alle Sprachen verfügbar sind²³⁾. Ferner gilt auch in diesem Zusammenhang, daß der entscheidende Faktor das behandelte Problem selbst ist, d.h. das optimal ausgewählte Werkzeug garantiert auch die beste Lösung der Aufgabe.

Die numerischen Fähigkeiten von KI-orientierten Sprachen können für ältere Dialekte ohne weiteres als "verkümmert" bezeichnet werden, da im Mittelpunkt der Entwurfsphilosophie eben die nicht-numerische Symbolverarbeitung stand. Mittlerweile gibt es allerdings z.B. LISP-Versionen, die eindeutig bessere Möglichkeiten zur Verfügung stellen, als ursprünglich numerisch orientierte Sprachen²³⁾. Die Interaktionsfähigkeit und Flexibilität bietet dem Benutzer einen hohen Komfort, so existieren Programmpakete, die z.B. hinsichtlich der numerischen Stabilität eine höhere Sicherheit garantieren als FORTRAN-Implementationen. PROLOG hinkt in dieser Beziehung etwas nach, was auch darauf zurückzuführen ist, daß LISP-Dialekte mit derartigen Fähigkeiten auf Spezial-Hardware implementiert sind.

Umgekehrt gilt für die Fähigkeiten prozeduraler Sprachen in der Verarbeitung symbolischer Ausdrücke, daß modernere Konzepte es durchaus zulassen, nicht-numerische Variablenwerte zu verarbeiten. Aber auch der zitierte FORTRAN-Programmierer wird (zurecht) den Standpunkt vertreten, daß mit "seinem" Werkzeug derartige Mechanismen zu realisieren sind. Es ist letztlich in dieser Hinsicht nur eine Frage des Komforts, den die ausgewählte Programmierungsumgebung bietet, wie aufwendig die Realisierung verschiedener Funktionen ist. Daß die Stärken prozeduraler Sprachen nicht auf dieser Ebene liegen, ist durch das jeweilige Entwurfskonzept selbst begründet.

23) Mittlerweile sind auf Personal Computern Implementationen sogenannter KI-Sprachen verfügbar, die auch die Möglichkeit arithmetischer Operationen bieten.

Insbesondere für Anwendungen im technischen Bereich tritt die Anforderung auf, periphere Hardware manipulieren zu können. Diese On-Line-Fähigkeit wird besonders prägnant, wenn zeitkritische Probleme, z.B. die Verarbeitung von Meßwerten in kurzen Zeiträumen, behandelt werden. Prozedurale Sprachen weisen hier den Vorteil auf, daß aufgrund des imperativen Lösungsmechanismus eindeutige Aussagen über den Rechenzeitbedarf getroffen werden können [BLO87], wogegen die unvorhersehbare Suchtiefe deklarativer Sprachen die Einbindung von Peripherie-Operationen erschwert. Dieses Merkmal ist dadurch gegeben, daß Echtzeit-Programmierung im allgemeinen auf mathematischen Modellen basiert, die in Expertensystem-Implementationen gewöhnlich nicht nachgebildet werden, da der Ansatz der Lösung auf Heuristiken gestützt ist. Auf modellbasierte Expertensysteme soll an dieser Stelle nicht näher eingegangen werden, da diese Problematik eine andere Qualität darstellt, in der Zukunft wird dieser Bereich, gerade aufgrund der gewachsenen Leistungsfähigkeit der dann verfügbaren Hardware, aber eine zunehmende Bedeutung bekommen, insbesondere hinsichtlich solcher Aufgaben, die eine Steigerung der technischen Autonomie erfordern.

Steht dem Systementwickler die Möglichkeit zur Verfügung, eine prozedurale Programmiersprache mit einer sogenannten KI-Sprache über eine geeignete Schnittstelle zu "verbinden", so lassen sich die Vorteile der komplementären Werkzeuge kombinieren. Es zeigt sich bei diesem Interfacing, daß insbesondere die Stärken prozeduraler Sprachen zur Leistungssteigerung des Gesamtsystems, z.B. bezüglich der Rechenzeit, einen wesentlichen Beitrag leisten können [POP87].

Es ist zusammenfassend zu bemerken, daß der Leistungsvergleich der Programmierwerkzeuge in der dargestellten Palette auf der abstrakten Ebene der reinen Sprachen objektiv nicht sinnvoll erscheint, da die jeweiligen Fähigkeiten und Stärken auf gegensätzlichen Entwurfsphilosophien basieren. Es ist vielmehr im wesentlichen vom behandelten Problem (und der angestrebten Lösung) abhängig, wie akzeptabel eine Implementation realisiert werden kann. Grundsätzlich läßt sich sagen, daß mit jeder der behandelten Programmiersprachen, wie anhand der einschlägigen Literatur auch nachweisbar ist, ein Expertensystem erstellt werden kann. Es ist letztlich immer auf die Frage des Komforts der Entwicklungsumgebung zurückzuführen²⁴⁾, wie aufwendig sich eine Realisierung darstellt.

24) Das betrifft auch die Frage, wie vertraut ein Programmierer mit einer Sprache ist.

Der Nachweis der "Machbarkeit" läßt sich auch damit begründen, daß Programmiersprachen im Sinne von mnemonischen Abstraktionsebenen klassifizierbar sind. So sind z.B. einige PROLOG-Implementationen in der Sprache C geschrieben, was darauf hinweist, daß der PROLOG-Ablaufmechanismus mittels der Zeigerprogrammierung in C realisiert ist. Der PROLOG-Anwender benötigt aber nicht die Kenntnis dieser Programmiertechnik, um seine Programme zu schreiben. Weiterhin gilt für höhere Programmiersprachen, die an moderneren Konzepten orientiert sind, daß sie häufig in anderen problemorientierten Sprachen implementiert sind²⁵⁾. Auch hier gilt, daß der Anwender der höheren mnemonischen Abstraktionsebene keine Kenntnis über die Realisierung z.B. des Pointer-Mechanismus benötigt. Diese Betrachtung läßt sich fortsetzen bis in die "maschinennaheste" Ebene, den Hexadezimalcode zur Programmierung zu verwenden, der aber wiederum nur eine "Kurzschreibweise" dafür darstellt, daß an ganz bestimmten Leitungen zur Ausführungszeit "Low" oder "High", bzw. 0 oder 5 Volt anliegen soll, was der "Hex-Programmierer" zwar in der Regel weiß, diese Kenntnis im Grunde genommen aber ebenfalls nicht benötigt. Es ist natürlich - schon aus ökonomischen Gründen - nicht realistisch, dazu überzugehen, wissensbasierte Systeme in Assembler²⁶⁾ oder sogar direktem Hexcode zu entwickeln. Die Betrachtung von Programmiersprachen anhand der mnemonischen Abstraktion beweist aber, daß die Implementierung von Expertensystemen nicht an bestimmte Werkzeuge gebunden ist. Die Philosophie des wissensbasierten Programmentwurfs und die entwickelte Methodik der Wissensrepräsentation führen allerdings dazu, Programmiersprachen wie LISP und PROLOG zu bevorzugen. Darauf wird im folgenden Abschnitt nochmals gesondert eingegangen.

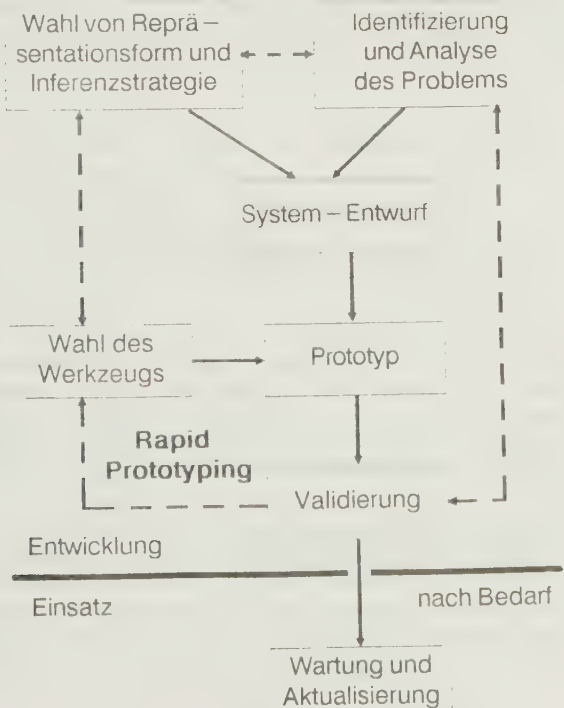
25) z.B. der (kommerziell vertriebene) COBOL-Compiler von Nixdorf ist in BASIC geschrieben.

26) Es existiert allerdings ein Expertensystem, das in Assembler geschrieben ist[EXP86].

2.3 Entwurfsschema für wissensbasierte Systeme

Bei Ansätzen zur Entwicklung von Expertensystemen [HAR86] läßt sich eine Unterscheidung hinsichtlich der Größe der behandelten Domäne treffen. So erfordern kleinere Probleme, etwa vom Umfang bis zu 50 Regeln, häufig keinen gezielten Entwurf der Lösung. Im Gegensatz z.B. zur strukturierten Programmentwicklung findet in der wissensbasierten Programmierung vielmehr dann die Technik Anwendung, die als Rapid Prototyping bezeichnet wird. Diese Lösungsfindung ist dadurch charakterisiert, daß kein "Entwurf auf dem Papier" vor der Implementation steht, sondern "direkt am Rechner" entwickelt wird. Für größere Domänen zeigt sich allerdings, daß auf die klassische Vorgehensweise der Konzepterstellung und der dann erst folgenden Umsetzung nicht verzichtet werden kann²⁷⁾. Dabei wird für den Expertensystem-Einsatz in

Bild 2.8 :
Entwurfsschema für
Expertensysteme



27) Eine Modularisierung der Aufgabe in kleinere Teilaufgaben führt allerdings wieder zur Rapid-Prototyping-Technik.

einer umfassenderen (betrieblichen) Umgebung sogar empfohlen, vor der Detailanalyse überhaupt erst das geeignete Problem auszuwählen. In Bild 2.8 ist ein Gesamtschema zur Systementwicklung dargestellt [MEY86].

2.3.1 Der innere Ablauf der Realisierung : Rapid Prototyping

Für die Entwicklung kleinerer Expertensysteme wird häufig empfohlen, die Auswahl des Werkzeugs an den Anfang der Problembearbeitung zu stellen. Damit wird in erster Linie die Verwendung von Shells propagiert, die dem Benutzer allein den Aufbau der Wissensbasis überlassen, wobei die Wissensrepräsentationsform meist festgelegt ist. Insbesondere aber unter dem Aspekt der Ausweitung des behandelten Problembereichs und dem damit verbundenen Übergang von kleinen Demonstrationssystemen zum einsatzfähigen Expertensystem(-Prototyp) steigt - wie bereits an früherer Stelle genannt - die Erfordernis, flexiblere Implementationen erstellen zu können. Die Restriktionen von (kleineren) Shells führen dann aber wieder auf die Verwendung reiner Programmiersprachen in der Evaluierung.

In der wissensbasierten Programmierung ist, bei geeigneter Modularisierung des Problems in kleinere Teilaufgaben, der Technik des Rapid Prototyping eine hohe Bedeutung beizumessen. In diesem Zusammenhang erscheint die Frage wichtig, warum gerade LISP und PROLOG sich so großer Beliebtheit erfreuen und zu den populärsten Werkzeugen in der Expertensystem-Entwicklung geworden sind. Bezüglich LISP spielt sicher auch Alter und insbesondere Herkunft der Sprache eine Rolle, der beiden gemeinsame objektive Grund ist aber das hohe Maß an Interaktivität dieser Sprachen, sowie die beiden eigene Programmstruktur.

In der prozeduralen Programmierung ist es erforderlich, Deklarationen der verwendeten Programmelemente²⁸⁾ durchzuführen. Eine solche Deklaration betrifft die Erfordernis eines Hauptprogramms. Dagegen ist in LISP jede Funktion zunächst eine selbständige Struktur für sich, in PROLOG jede Prozedur ein unabhängiges Programm, erst die Verschachtelung solcher Elemente

28) Unter Programmelementen seien hier sämtliche Programm- und Datenstrukturen verstanden.

führt dazu, daß Abhängigkeiten zwischen Programmteilen entstehen können. Ein Hauptprogramm im Sinne der prozeduralen Programmierung ist nicht vorhanden, wobei in LISP noch zusätzlich die Tatsache zu berücksichtigen ist, daß zwischen Programmen und Daten erst zum Zeitpunkt der Auswertung eines Ausdrucks unterschieden werden kann. Ferner ermöglichen beide Sprachen die Änderung von Programmen bzw. Programmteilen zur Laufzeit, die in prozeduralen Sprachen in der Regel eine Neuprogrammierung und Recompileation erfordert. Diese Eigenarten machen LISP und PROLOG zu den schnellsten Entwicklungswerkzeugen für das Rapid Prototyping.

Ein weiterer Aspekt, der hierbei zu berücksichtigen ist, ist das Charakteristikum wissensbasierter Programmierung, Mehrdeutigkeiten bzw. alternative Lösungen zuzulassen. Validationsformalismen, wie z.B. der bekannte Tabellentest bei algorithmenbasierten Programmen²⁹⁾, sind daher nur sehr schwierig anzuwenden. Das Verfahren des "schnellen Probierens" erweist sich hier hinsichtlich der Möglichkeit des unabhängigen Testens kleiner Programmteile von LISP- oder PROLOG-Programmen als effektiver, da die Interaktivität der Sprachen die schnelle Fehlerkorrektur erleichtert.

2.3.2 Der äußere Ablauf der Realisierung : Systementwurf auf der Basis einer erschöpfenden Problemanalyse

Von kleinen Demonstrationssystemen abgesehen, die aufgrund geringer Komplexität der Aufgabe nur eine oberflächliche Problem-betrachtung erfordern, liegt die Hauptarbeit des Expertensystem-Entwurfs sicherlich in der sorgfältigen und detaillierten Analyse der Domäne und der folgerichtigen Interpretation der bestehenden Zusammenhänge. Diese Aussage soll keine Minderung der Qualität von Demonstrationssystemen darstellen, denkt man jedoch an die gängigen Beispiele aus der Literatur [IUD84], z.B. das Auto, das nicht anspringt, und durch ein etwa 30 Regeln umfassendes System (erfolgreich ?) diagnostiziert wird, so stellt sich hier nicht unbedingt ein Problem, das einen absoluten Experten erfordert. Der Sinn solcher Demosysteme ist wohl

29) Für eine größere Zahl von Variablen in einem Programm bekanntlich sehr beschwerlich durchzuführen.

darin zu sehen, die Anwendbarkeit wissensbasierter Systeme für derartige Aufgaben zu verifizieren. Eine gewisse Vorarbeit in der Domänen-Analyse ist natürlich auch für diese Rapid-Prototyping-Systeme zu leisten, da die Validierung davon abhängig ist.

Den ersten Schritt zum Systementwurf "echter" Expertensysteme, sieht man einmal von der Empfehlung der Auswahl des geeigneten Problems ab, stellt somit die Analyse und Interpretation der Aufgabe dar, also die Gewinnung des Wissens über die Domäne und die Überführung dieses Wissens in eine formale Beschreibung. Die Aspekte dieser arbeitsaufwendigsten Tätigkeit der Expertensystem-Erstellung werden in Abschnitt 3 diskutiert. Die Vorgehensweise bei der Akquirierung des Wissens kann dabei bereits, wie in Bild 2.8 angedeutet, durch die Wahl von Wissensrepräsentationsform und Inferenzstrategie beeinflusst sein - auch in Hinblick auf die Auswahl eines bestimmten Werkzeugs - da, wie bereits in Abschnitt 2.2 ausgeführt, z.B. pragmatische Gründe für die Auswahl einer Implementierungssprache maßgeblich sein können. Auch sei hier noch erwähnt, daß es z.B. in Verbundprojekten üblich ist, derartige Rahmenbedingungen zuvor festzulegen, da die Teilmodule letztlich kompatibel sein müssen.

Der Wissensgewinnung und -klassifizierung folgt dann die Formalisierung unter Beachtung der gewählten Repräsentationsform und Ablaufstrategie im Systementwurf. Dabei kann, umgekehrt zum vorher genannten, auch die Problemanalyse zur bestgeeigneten Repräsentationsform führen, sowie ferner die Auswahl des Werkzeugs beeinflussen. Ist der Systementwurf an der Stärke wissensbasierter Programmierung orientiert, d.h. modular in Teilaufgaben zergliedert, können im inneren Ablauf der Realisierung die Vorteile des Rapid Prototyping genutzt werden. Allerdings ist die Vorgehensweise dabei an die gewählte Implementierungssprache gebunden, denn in dem Fall, daß eine prozedurale Sprache zur Realisierung gewählt wird, empfiehlt sich ein detaillierter Entwurf, z.B. im Sinne strukturierter Programmentwicklung, der gerade bei der Behandlung zeitkritischer Aufgaben relevant ist.

2.3.3 Rückkopplungen der Realisierung : Validation des Systems

Nach der Wissensakquisition ist in der Aufgabe der Validierung des entwickelten Prototyps der zweite zentrale Schwerpunkt der Systemevaluierung zu sehen. Dabei wird es in einer ersten Phase wichtiger sein, die Abläufe des Programms als schlüssig zu verifizieren, denn evtl. fehlendes Wissen zu ergänzen, was in eine zweite Phase - möglichst unter Hinzuziehung des befragten Experten - verlegt werden kann.

Die Determinierung des Inferenzmechanismus' widerspricht dabei nicht dem Prinzip der nicht-deterministischen Lösungsfindung, da nicht gefordert werden muß, Alternativen bzw. Mehrdeutigkeiten auszuschliessen. Dieses Merkmal der wissensbasierten Programmentwicklung sollte erhalten bleiben, sofern sinnvolle Mehrdeutigkeiten die Domäne charakterisieren. Es ist vielmehr erforderlich, zu gewährleisten, daß, wenn zwei Konsultationen mit den gleichen Eingangsdaten durchgeführt werden, diese auch gleiche Ergebnisse liefern³⁰⁾. Obwohl die Charakteristik der alternativen Lösung von semantischer Bedeutung für die Lösungsfindung ist, steht im Vordergrund dieser Validierungsphase die syntaktische Überprüfung und die Verifizierung des Programmablaufs. Die Berücksichtigung z.B. von Ergebnissen früherer Konsultationen, in Diagnosesystemen realisierbar durch eine mitzuführende Statistik, die die Beeinflussung von Gewichtungsfaktoren bewirkt, sollte in der ersten Phase der Validation (noch) nicht erfolgen, da erst die Verfeinerung des Wissens Aufschluß darüber zuläßt, ob diese Möglichkeit für das betrachtete Problem sinnvoll einzusetzen ist.

In der zweiten Phase der Systemvalidierung nimmt die Verbesserung und Verfeinerung des repräsentierten Wissens den entscheidenden Anteil ein, wobei der befragte Experte hinzugezogen werden sollte. Diese Rückkopplung ist als "iterativer Prozeß" aufzufassen, der mehrmals durchgeführt werden kann. Anfangs wird es dabei wesentlich sein, Widersprüche und Inkonsistenzen im Wissen zu erkennen. In einer späteren Iteration ist es dann möglich, die Optimierung der Kontrollstrategie und des Regelwerks vorzunehmen.

Im ungünstigen Fall können die Ergebnisse der zweiten Phase der Validation darauf führen, daß das entwickelte System nicht den gestellten Anforderungen entspricht. Kann das gewählte Werkzeug

30) Der Non-Determinismus wissensbasierter Systeme darf nicht zu "zufälligen" Ergebnissen führen.

als Ursache erkannt werden, so ist eine neue Prototyp-Implementierung unter Umständen auf der Basis des bestehenden formalen Systementwurfs möglich. Derartige Effekte könnten z.B. durch das Laufzeitverhalten einer Sprache bzw. eines speziellen Sprachdialekts bestimmt sein. Ist die mangelhafte Leistungsfähigkeit dagegen auf die gewählte Wissensrepräsentationsform zurückzuführen, was ebenfalls mit dem gewählten Werkzeug zusammenhängen kann, z.B. bei Verwendung eines Rahmensystems, so ist der Systementwurf auf der Basis einer besser geeigneten Repräsentationsform zu modifizieren. Im Extremfall kann ein völlig neuer Systementwurf, basierend auf einer verbesserten Vorgehensweise bei der Wissensakquisition³¹⁾, erforderlich werden. In gleicher Weise gilt dies für eine falsch gewählte Kontrollstrategie [PUP86].

31) Aufgrund der Vorgehensweise bei der Wissensakquisition, z.B. durch die vom Knowledge Engineer gewählte Art der Fragestellung an den Experten, kann sich ergeben, daß die formale Struktur des Wissens realisierungstechnisch nicht (effektiv) umzusetzen ist [NOE85].

3. Probleme der Wissensakquisition und -umsetzung

Die wissensorientierte Programmierung findet zwar zunehmend auch für solche Aufgaben Verwendung, bei denen modellbasierte Beschreibungen den Formalismus der Problemlösung charakterisieren, sie stützt sich aber ursprünglich auf das Prinzip, nicht den Lösungsweg selbst, sondern die verfügbaren "Daten" zu definieren und damit eine Lösung zu generieren.

In der algorithmenorientierten Programmierung ist dagegen die Vorgehensweise eindeutig festgelegt. So werden dort Problem und Lösung, z.B. in Form eines mathematischen Modells, das eine hinreichend genaue Abbildung der Wirklichkeit wiedergibt, in der jeweiligen Programmiersprache fest implementiert und sodann aufgrund von Eingangsdaten Ausgangsdaten erzeugt. Die "richtige" Bedeutung dieser Daten, z.B. die Erfüllung von Kausalitätsbedingungen, ist dabei ohne Belang, ein solches Rechnerprogramm "kann" z.B. auch negative Massen verarbeiten. Es ist vielmehr die Aufgabe des Systementwicklers, den Gültigkeitsbereich des implementierten Algorithmus' zu validieren und festzulegen. Dabei bezieht sich die Einschränkung "hinreichend genaue Abbildung" auf die Tatsache, daß jedes mathematische Modell nur eine eingegrenzte Darstellung der Wirklichkeit liefern wird¹⁾,

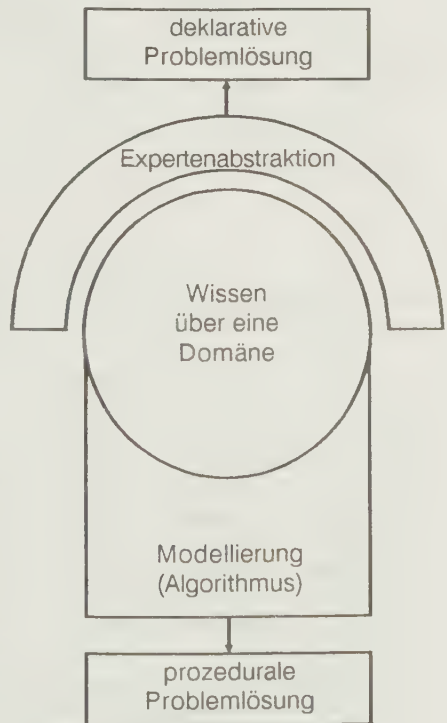


Bild 3.1 : Ansätze der Wissensmodellierung

1) z.B. die aus der Systemtheorie bekannte Annahme, lineare Modelle n-ter Ordnung aufzustellen, stellt schon eine Einschränkung dar, da jedes reale physikalische System nichtlinear und unendlicher Ordnung ist.

durch die Vernachlässigung von Einflüssen geringer Bedeutung aber eine hinreichend genaue Beschreibung sein kann.

Im Gegensatz dazu stützt sich die wissensbasierte Problembeschreibung auf die Definition von Fakten und Relationen über Objekte, die die Heuristiken eines menschlichen Experten über die Domäne wiedergeben. Zentrale Frage der Aufgabenbeschreibung ist nicht, wie programmiert wird, sondern was man programmiert. Dieser Ansatz erfordert, daß die beschriebenen Tatsachen und Zusammenhänge nicht nur formal syntaktisch, sondern insbesondere auch semantisch vollständig und korrekt sind, da die hierbei bearbeiteten "Daten", eben das Expertenwissen, im Mittelpunkt der Problemlösung stehen. Die Akquirierung dieses Expertenwissens, das einer Abstraktion der Domänencharakteristik in das Wissensmodell des Experten entspricht, durch den Knowledge Engineer, ist die bedeutsamste Aufgabe in der Entwicklung wissensbasierter Systeme. Die prinzipielle Vorgehensweise soll im folgenden Abschnitt am Beispiel des Ablaufs der Knowledge-Engineering-Phase [NOE85] behandelt werden.

3.1 Prinzipieller Ablauf der Knowledge-Engineering-Phase

In Bild 3.2 ist dargestellt, welche Anteile von Wissen über einen Problembereich grundsätzlich zu der Fähigkeit beitragen, eine Aufgabe wie ein Experte dieser Domäne zu lösen. Unter der Beschreibung der Domäne ist dabei das gesamte vorliegende Wissen über das betrachtete Problem zu verstehen, welches im Verlauf des Knowledge-Engineering-Prozesses einer "Transformation" in eine programmiertechnisch realisierbare formale Darstellung zu unterziehen ist. Für die Formalisierung von Expertenaussagen durch den Entwickler ist es dabei erforderlich, die richtige Deutung des akquirierten Wissens zu gewährleisten. Das betrifft in erster Linie die Frage des Vokabulars, d.h. die Festlegung einer gemeinsamen Terminologie zwischen Experte und Entwickler. Zu berücksichtigen ist dabei, daß ein Experte zu einem Fachmann (einem anderen Experten) anders sprechen wird als zu einem Laien. Erst auf der Basis einer gemeinsamen Nomenklatur ist der Knowledge Engineer in der Lage, das Problem im Sinne des Experten einzuschätzen und richtig zu formalisieren.

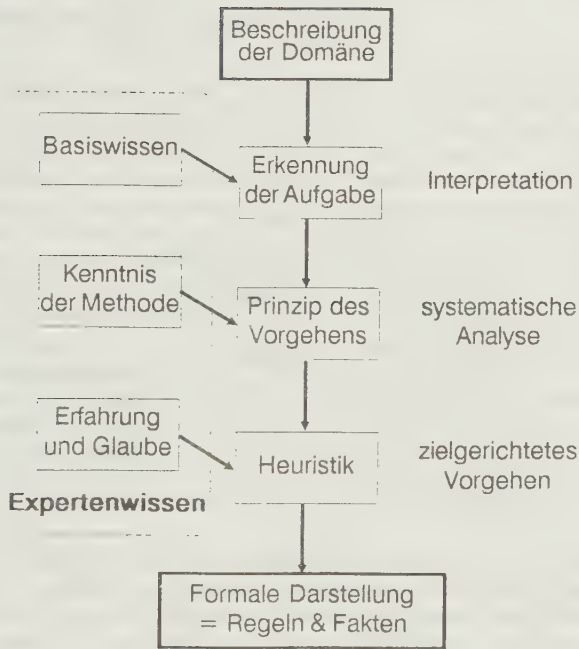


Bild 3.2 : Ablauf der Wissensakquisition

Die erste Aufgabe des Wissensingenieurs - vor einer Experten-konsultation - wird es sein, sich mit dem gestellten Problem vertraut zu machen, um gezielt und richtig Fragen an den Experten richten zu können, die tiefere und detailliertere Aspekte der Domänenbeschreibung "aufdecken". Die Gewinnung dieses Basiswissens erfolgt in der Regel nicht durch Experteninter-view, sondern ist durch Literaturrecherchen, Sichtung und Auswertung von Datenblättern und Pflichtenheften, etc. zu leisten. Durch dieses Grundwissen ist der Knowledge Engineer dann in der Lage, das Problem zu definieren und eine Interpretation der gestellten Aufgabe hinsichtlich des Lösungsansatzes durchzuführen.

Die Kenntnis der Methodik einer Lösungsfindung ist der zweite Wissensanteil, der zur Problemlösung beiträgt. Unter Hinzunahme dieser Wissenskategorie zum Basiswissen ist ein "Verwender" dieses Wissens in der Lage, die prinzipielle Vorgehensweise zu

beschreiben und eine systematische Analyse durchzuführen. Dabei gilt auch für das Methodenwissen, daß es sich der Knowledge Engineer - zumindest zum Teil - unabhängig vom Experten verschaffen kann (z.B. durch adäquate Lehrbücher). So sollte z.B. jeder Elektroingenieur dazu befähigt sein, eine elektronische Schaltung systematisch zu diagnostizieren, ohne einen Experten der speziellen Schaltung konsultieren zu müssen²⁾.

Der wichtigste Anteil des Expertenwissens, der bei der Systementwicklung zu berücksichtigen ist, ist der Bereich derjenigen Kenntnisse, die sich auf die - oft langjährige - Erfahrung eines Experten mit dem ausgewählten Problem, und die aus dieser Erfahrung gewonnenene intuitive Vorgehensweise bei der Problemlösung stützen. Erst durch die Einbeziehung dieser Wissenskategorie in den Systementwurf ist gewährleistet, daß die Definition eines Expertensystems, als "ein Rechnerprogramm, daß die Fähigkeiten eines menschlichen Experten bis zu einem gewissen Grad nachbildet" [HAY83], erfüllt wird. Damit ist dann die Fähigkeit erlangt, vom systematischen Vorgehen gelöst die Lösungsfindung aufgrund bestimmter Symptom- oder Situationsmuster zielgerichtet durchzuführen.

Das dargestellte Schema der Wissensgewinnung gemäß Bild 3.2 ist nicht zwingend als sequentieller Ablauf aufzufassen, da z.B. in der Phase des Experteninterviews Aspekte der verschiedenen Wissensanteile gemeinsam bestimmend für ein Charakteristikum der Domäne sein können. Weiterhin hängt der "Ansatzpunkt" der Akquirierung im Einzelfall vom Kenntnisstand des Knowledge Engineers über das betrachtete Problem und dessen Umfeld ab. Letztlich ist noch zu berücksichtigen, daß sich das Expertenwissen zwar generell aus den drei genannten Kategorien zusammensetzt, jedoch die wenigsten Experten lückenlose Kenntnisse nachweisen können. So gilt gerade für komplexere Domänen, daß mehrere Experten in der Wissensakquisition konsultiert werden müssen, um alle charakteristischen Merkmale der Domäne zu erfassen und die bestmögliche formale Beschreibung des behandelten Problems zu gewinnen. Für die später behandelte Überwachungsaufgabe chemischer Reaktoren stellt sich z.B. die Aufgabe, das Wissen von verschiedenen Experten unterschiedlicher Disziplinen und unterschiedlichen Ausbildungsniveaus (Wartenfahrer, Chemiker, Ingenieure) zu akquirieren, wobei zusätzlich eine vorgegebene konzeptionelle Methodik zu beachten ist [SCM86].

2) Im Einzelfall, d.h. abhängig von der jeweiligen Funktion und der Komplexität einer Schaltung, können dieser Befähigung natürlich Grenzen gesetzt sein.

3.2 Probleme der Wissensverarbeitung

Bei der Akquisition und Formalisierung des Domänenwissens stellt sich dem Knowledge Engineer die Aufgabe, eine streng abgegrenzte, aber in sich vollständige Beschreibung des betrachteten Problems aufzustellen. Insbesondere in dem Fall, daß mehrere Experten zur Problemlösung beitragen, kann es jedoch erforderlich werden, inkonsistente Aussagen zu klassifizieren, sowie Wissen aus Randbereichen der Domäne und Allgemeinwissen zu berücksichtigen. Die wesentlichen Aspekte dieser Wissensverarbeitungsproblematik sollen im folgenden diskutiert werden.

3.2.1 Das Expertenproblem

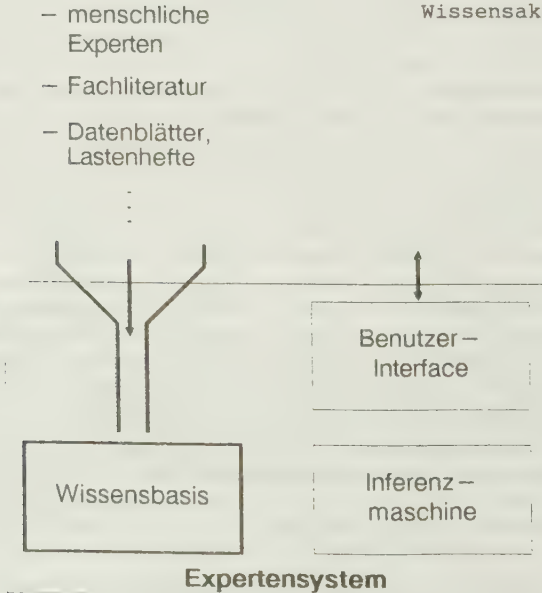
Die Ausstellung von Wissensbasen für kleine und mittlere Probleme, etwa im Umfang 30 - 150 Regeln, erfordert (im Normalfall) nicht den Entwicklungsaufwand, der der Wissensakquisition den Ruf der zeitaufwendigsten Tätigkeit der Expertensystem-Erstellung eingebracht hat. Der Systementwickler hat bei derartigen Problemen in der Regel das Wissen eines einzelnen Experten aufzubereiten, sofern er sich nicht ohnehin nur auf Literaturquellen, Datenblätter u.ä. stützt. Der Aufwand der Wissensgewinnung steigt allerdings mit zunehmender Komplexität der Domäne (bis zu einem gewissen problemabhängigen Maß) exponentiell. So wird z.B. für die Entwicklung der Wissensbasis von DENDRAL, einem der ältesten Expertensysteme, das zur Analyse von Molekularstrukturen in der Chemie eingesetzt wird, der Zeitaufwand auf ca. 15 Mannjahre geschätzt [LIN80].

Mit der Evaluierung der Wissensbasis einer komplexeren Aufgabe ist ferner meist verbunden, daß mehrere menschliche Experten zur Charakterisierung der Domäne und zur Lösungsfindung beitragen. Für den Knowledge Engineer ergeben sich damit im wesentlichen drei dominierende Problembereiche. Der erste Schwerpunkt ist generell die Aufgabe, die Vielfalt an akquirierter Information in das gewählte Repräsentationsschema einzupassen. Daran schließt sich an, das gewonnene Wissen unter Konsistenzbedingungen zu betrachten, um Widerspruchsfreiheit und Vollständig-

keit zu gewährleisten. Ferner besteht die Anforderung, Akzeptanz bezüglich des formalisierten Wissens bei den Experten herzustellen, um auch in der folgenden Phase der Weiterentwicklung vom Prototyp zum Produktionssystem die "Mitarbeit" der Experten sicherzustellen.

Der Aufbau einer Wissensbasis bedeutet, ein Programm aus einer Fülle von "unsortierten" Informationen, dem Wissen der Experten, zu erstellen, ohne die Möglichkeit nutzen zu können, externe Datenträger dieses Wissens (z.B. Tonbänder von Experteninterviews) an den Zielrechner zu koppeln. Zunächst ergibt sich damit das grundsätzliche Problem, mit den eingeschränkten Eingabemöglichkeiten diese umfangreiche Aufgabe zu bewältigen. Das wesentlichere Merkmal dieser, in der Literatur als Bottle-Neck-Problem bekannten Schwierigkeit [CES87], besteht allerdings in der Strukturierung und "Filterung" des Wissens in relevante und unbedeutende Bestandteile. Wie im vorigen Abschnitt beschrieben, wird sich der Knowledge Engineer in einer ersten Phase der Wissensakquisition mit dem behandelten Problem soweit vertraut machen müssen, um mit einem Experten für die Domäne - auf der Basis einer gemeinsamen Terminologie - die Analyse der Domänen-

Bild 3.3 : Bottle-Neck-Problem der Wissensakquisition



charakteristik im Detail durchführen zu können. Sind mehrere Experten an der Akquirierung beteiligt, besteht unter Umständen das Problem, daß keine "einheitliche" Sprachregelung getroffen werden kann, und der Wissensingenieur zu einer Art "Selektion" der ihm zugetragenen Informationen gezwungen wird, um eine formale Darstellung des Wissens zu ermöglichen. Zusätzlich erschwert wird diese Aufgabe, wenn zwischen den beteiligten Experten ein "hierarchisches Verhältnis" besteht, z.B. die (verbale) Interpretation eines Chemikers gegen die eines Angehörigen des Leitwartenpersonals steht.

Die folgende Konsistenz- und Widerspruchsprüfung des akquirierten Wissens ist bereits die erste Iteration des weiteren Entwicklungsprozesses der Wissensbasis. Daher sollten in diese Phase die beteiligten Experten einbezogen werden. Das Hauptproblem wird es sein, inhaltliche Differenzen (zwischen den Expertenmeinungen) auszuräumen, um zu einer semantisch eindeutigen Beschreibung zu gelangen. Dabei kann die (inhaltliche) Interpretation einer Problematik wiederum entscheidend für das Auftreten von Widersprüchlichkeiten sein, was folgendes triviale, von der Betrachtung technischer Systeme gelöstes Beispiel verdeutlicht : werden Staatsmänner der Vergangenheit von Historikern beurteilt, so sind häufig Nebenbedingungen, wie die aktuelle politische Situation, dafür bestimmend, wie deren Leistungen einzuordnen sind.

Die Beseitigung inhaltlicher Widersprüche und Inkonsistenzen trägt dazu bei, das "Interesse" der Experten an der weiteren Verfeinerung und Optimierung der Wissensbasis zu erhalten. Dabei ist zu berücksichtigen, daß das Expertensystem für die Experten in der ersten oberflächlichen Betrachtung als ein Ersatz für ihre Tätigkeit angesehen werden könnte. Dem Knowledge Engineer fällt somit die Aufgabe zu, den Experten den Sinn und die Notwendigkeit des Expertensystems plausibel zu begründen, um den weiterführenden iterativen Entwicklungsprozeß in Gang zu setzen. Entscheidend für die Akzeptanz der Experten dürfte dabei sein, daß sie in dem "Verhalten" des Systems eine tatsächliche Abbildung ihres Wissens erkennen.

3.2.2 Konsistenz und semantische Bedeutung einer Aussage

Bei der Beurteilung von Expertenwissen ist der Systementwickler stark an die Frage gebunden, inwieweit die dargestellten Zusammenhänge relevant für die Problemlösung sind. Dabei kann es zu Widersprüchen zwischen der Konsistenz einer Aussage und ihrer semantischen Bedeutung im Fall bestimmter Instanzen kommen. Ein Beispiel möge dies im folgenden verdeutlichen :

Ein (z.B. framebasiertes) Expertensystem zur Beratung von Innenarchitekten enthält für die Hauptklasse "Möbel" eine Klasse "Stuhl". Ein Stuhl sei so definiert, daß er a) vier Beine, b) eine Sitzfläche und c) eine Lehne hat. Diese Attribute können mit Nominaleninträgen versehene Slots des definierenden Frames sein. Weitere Slots des Frames "Stuhl" könnten Farbe, verwendetes Material, Stilrichtung, usw. sein. Die obige Definition stellt allerdings eine inkonsistente Aussage dar, weil z.B. nicht alle Stühle vier Beine haben (man denke an Bürostühle !). Andererseits ist die Aussage semantisch korrekt, da für bestimmte Instanzen (Stühle) die Definition zutrifft. Die Vollständigkeit der Definition ist in diesem Beispiel abhängig von den einzelnen Instanzen der Klasse "Stuhl".

Für den Knowledge Engineer stellt sich somit die Frage, inwieweit derartige, als Nominalwerte repräsentierte Eigenschaften sinnvoll für die Lösung einer Aufgabe sind, und wie zu verfahren ist, um eine möglichst vollständige Definition des Objekts zu gewährleisten. Dabei ist die korrekte und effiziente Verarbeitung der Objektdefinition außerdem eng an die gewählte Wissensrepräsentationsform gebunden. Die kompakte Darstellung in Form einer Frame-Definition erscheint für das genannte Beispiel am ehesten die Forderung einer vollständigen Objektbeschreibung erfüllen zu können. Eine vergleichbare Repräsentation z.B. in Form von Produktionsregeln würde einen beträchtlich höheren Aufwand erfordern. Deutlicher wird dieses Konsistenz-Problem noch, wenn man statt eines Beratungssystems ein 3D-Bildverarbeitungssystem betrachtet [TOL87]. Die Objekterkennung wäre mit obiger Definition an strenge Restriktionen gebunden.

Die Umsetzung von Wissen in einem Programm ist also in bezug auf korrekte Darstellung und Gewährleistung der richtigen Verarbeitung eng an die Konsistenz einer Aussage gebunden. Damit verbunden ist aber die Frage der Relevanz des repräsentierten Wissens. Eine Domäne soll zwar möglichst vollständig und gegen "Allgemeinwissen" abgegrenzt beschrieben werden, jedoch ist die

Elimination von Sonderfällen, die im Rahmen der Problemlösung irrelevant sind, mitunter wünschenswert, um den Aufgabenbereich einzuschränken. Auf obiges Beispiel bezogen : Ein Beratungssystem für "Wohnzimmer-Innenarchitekten" würde nicht das Wissen über das typische Büromöbel "fünfstrahliges Drehstuhl" benötigen³⁾.

Nicht konsistentes Wissen kann andererseits auch als vages Wissen aufgefaßt werden. So kann damit in gleicher Weise verfahren werden, wie mit dem Wissen, das der Experte von sich aus nicht in die Kategorien "mit Sicherheit wahr" und "mit Sicherheit falsch" einordnen kann. Die Belegung von Aussagen mit Sicherheitsfaktoren kann hier in bezug auf die Evaluierung von Problemlösungen über Zwischenergebnisse die Effizienz der Konfliktlösungsstrategie erhöhen, bzw. ferner auch die Möglichkeit offenhalten, nicht aufgrund einer als sicher eingestuften Aussage den Lösungsweg abzubrechen [BAR81], [COH81].

Ein Beispiel dieser Verfahrungsweise ist der Umgang mit Wissen in MYCIN [BUC84]. In diesem System werden Diagnosen und Hypothesen grundsätzlich mit einer bestimmten Gewißheit repräsentiert, da (unmittelbar) in den Regeln nur einige wenige Symptome die Bedingung für das Zutreffen einer Konklusion bilden. Diese Repräsentationsform ist beispielhaft für den Umgang mit Domänen großer Komplexität.

Im zuvor aufgeführten Beispiel würde die Vergabe eines Konfidenzfaktors in der Form "ein Stuhl hat wahrscheinlich vier Beine" zu einer Prüfung der evtl. vorliegenden Ausnahmen führen. Eine Verbesserung des Systemverhaltens ist hier erreichbar durch die Anwendung von Kontrollstrategien, die die vergebenen Konfidenzfaktoren aufgrund früherer Konsultationsergebnisse beeinflussen⁴⁾. Derartige variable Sicherheitsfaktoren können zur zielgerichteteren Lösungsfindung beitragen, z.B. in Diagnosesystemen für ein bestimmtes technisches System, um mit der Zeit offensichtliche Fertigungsfehler schneller zu erkennen. Jedoch birgt ein solcher Repräsentationsansatz auch die Gefahr einer falschen Bewertung, wenn z.B. Fertigungsfehler bestimmter Bauteilserien zur Erhöhung der Sicherheitsfaktoren führten, und die Folgeserie dieses Fehlverhalten nicht mehr aufweist. Es ist im Einzelfall also zu prüfen, ob die Verwendung variabler Konfidenzfaktoren für die Problemlösung sinnvoll einzusetzen ist. Nochmals auf obiges "Architekten-System" bezogen : Eine individuelle Bewertung der Eigenschaften des Objekts "Stuhl" für den einzelnen Architekten, der ja seinen eigenen Stil kreiert,

3) Domänenbegrenzung aufgrund von Konsistenzbedingungen mit Hilfe des Prinzips "was ich nicht sehen will, das existiert auch nicht".

4) z.B. durch die Berücksichtigung einer Statistik.

könnte eine schnellere und damit effizientere Verarbeitung der Ausnahmenprüfung bewirken, andererseits aber auch den Schlußfolgerungsmechanismus negativ beeinflussen, wenn der Architekt seinen Stil, in diesem Fall z.B. die Anzahl der Beine eines Stuhles, wechselt.

Hinter das Konsistenz- und Semantikproblem tritt die Forderung nach syntaktisch korrekter Zuordnung von Expertenaussagen eher in den Hintergrund. Das ist u.a. auch darauf zurückzuführen, daß spezielle Tools zur Realisierung von Expertensystemen nicht so umfangreiche Vorschriften für Daten- und Kontrollstrukturen aufweisen, wie herkömmliche imperative Programmierwerkzeuge. Die formale Richtigkeit der Beschreibung von Wissen ist vielmehr an die gewählte Wissensrepräsentationsform gebunden und liegt somit hauptsächlich in der Verantwortung des Knowledge Engineers. Der Experte kann zu diesem Aspekt des Repräsentationsproblems nur dann beitragen, wenn er ausreichende Kenntnisse in der Expertensystem-Entwicklung aufweist. Die Einhaltung von formalen Vorschriften⁵⁾ tritt aber aus den oben genannten Gründen ohnehin nicht in dem Maße auf wie in der algorithmenbasierten Programmentwicklung⁶⁾.

3.2.3 Randwertprobleme, Domänenbegrenzung und Verallgemeinerung spezieller Aussagen

Unter Randwerten einer Domäne sollen hier nicht solche Aussagen verstanden werden, die im Sinne einer effizienten Problemlösung nicht berücksichtigt werden (können), obwohl sie zur Domäne gehören, wie in o.a. Beispiel die Vernachlässigung eines bestimmten Typs "Stuhl". Es sollen vielmehr diejenigen Sonderfälle von Expertenwissen betrachtet werden, die für eine genügende Problemlösung erforderlich sind, sowie der Anteil des Wissens, der gegen Allgemeinwissen nicht eindeutig abgrenzbar ist. Das Problem, derartiges Wissen in der Repräsentation in geeigneter Form zu erfassen, ist eine der wichtigsten Aufgaben des Knowledge Engineers.

5) das betrifft z.B. Sprachelemente, Datenstrukturen, Rekursionsmechanismen, u.ä.

6) Nachteil dieses scheinbaren Vorteils ist allerdings die mangelhafte Software-Dokumentation von KI-Programmen.

Spezielles Wissen in Randzonen einer Domäne kann dazu beitragen, mögliche "Sackgassen" einer Problemlösung zu vermeiden, birgt aber andererseits auch die Gefahr, solche Sackgassen zu schaffen.

Insbesondere in der Medizin ist derartiges Randwissen anzutreffen, wenn man an Mediziner denkt, die Spezialisten für ein einziges Organ sind. Mir ist z.B. ein Fall bekannt, wo ein Patient auf Nierenfunktionsstörungen behandelt wurde, später auch mögliche Wechselwirkungen zwischen Nieren und Magen-Darm-Trakt in Betracht gezogen wurden, der Patient jedoch rätselhafterweise auf keine Therapie ansprach. Ein Expertensystem, das auf der Grundlage des Wissens der behandelnden Ärzte entwickelt worden wäre, hätte in diesem Fall seine Grenzen gefunden, obwohl es sich bei einigen dieser Mediziner um Internisten, also Fachleuten für das Problem gehandelt hat. Der (nur widerwillig !) hinzugezogene Nierenspezialist konnte die Ursache der Krankheit tatsächlich lokalisieren : Er ordnete die Entfernung der vergrößerten Mandeln des Patienten an.

Dieses Beispiel zeigt, daß hochspezialisiertes Wissen eines Experten in gewissen Fällen notwendig ist, um Problemlösungen in Randzonen einer Domäne zu generieren. Gerade für Anwendungen in der inneren Medizin gilt, daß die Abgrenzung der betrachteten Domäne kritisch werden kann, wenn bestimmte Klassen von inneren Krankheiten als Problem bearbeitet werden, da die tatsächliche Domäne eben die gesamte innere Medizin ist⁷⁾. Diese Charakteristik ist durchaus übertragbar auf technische Anwendungen, wenn Expertensysteme für Teilbereiche eines umfangreicheren Problems eingesetzt werden. Mögliche Wechselwirkungen zwischen Teilprozessen, die nicht in der Repräsentation erfaßt werden, können Validitätsbereich und Leistungsfähigkeit eines Expertensystems zu seinem Nachteil beeinflussen.

Die Abgrenzung des reinen Domänenwissens gegen Allgemeinwissen ist ein anderer Aspekt der Randwertproblematik. Dabei stellt sich die Frage, inwieweit allgemeines Wissen, z.B. die Kenntnis bestimmter Kausalzusammenhänge, als Bestandteil des domänenspezifischen Wissens repräsentiert werden muß. Desgleichen kann Wissen über das Umfeld eines Problembereichs für die Lösungsfindung von Bedeutung sein.

Grundsätzlich gilt für den Entwurf wissensbasierter Systeme, daß die Definition der Nachbildung menschlicher Expertise nur in einem streng abgegrenzten Anwendungsbereich zu erfüllen ist,

7) Aus diesem Grunde werden seriöse diagnostische Expertensysteme, wie z.B. das im klinischen Einsatz befindliche PUFF [AIK84], auch nur zur Beratung und Unterstützung des Arztes eingesetzt.

was insbesondere auch auf die technischen Beschränkungen der Realisierbarkeit zurückzuführen ist (Speicherplatzbedarf, Rechenzeit). Besteht allerdings die Notwendigkeit, allgemeingültigeres Wissen in die Problemlösung einzubeziehen, so muß dafür gefordert werden, daß auch bezüglich dieser zu repräsentierenden Zusammenhänge eine strenge Abgrenzung gegen das Problem nicht tangierende "Seiteneffekte" sichergestellt ist, um eine kombinatorische Explosion auszuschliessen.

Insbesondere die Einbeziehung von Deep-Reasoning-Prinzipien [DAV84], [MOZ87] in Diagnosesysteme stellt hinsichtlich dieses "Aufbläh-Effekts" ein Problem für den Systementwurf dar, da er auf der Verallgemeinerung von Beschreibungsmechanismen einerseits und der Hinzunahme von - diese Verallgemeinerung berücksichtigenden - zusätzlichen Regeln andererseits basiert.

Für die Aufgabe der Fehlersuche an elektronischen Schaltungen, die (sofern möglich) bis auf Bauteilebene eine Diagnose liefern soll, gilt z.B., daß ein Entwurf auf der Basis der Heuristik eines Experten für eine bestimmte Schaltung zu einem akzeptablen Expertensystem führt [HOF87]⁸⁾. Die Einbeziehung allgemeiner Regeln über Bauteile in bestimmter technischer Ausführung, z.B. IC's in CMOS- oder TTL-Technologie etc., erfordert aber für jede zu berücksichtigende Technologie eine eigene Regelgruppe, um aufgrund dieses allgemeineren Wissens zu einer Diagnose zu gelangen.

Im Sinne des heuristisch basierten Entwurfs ist die jeweilige Herstellungstechnik irrelevant für die Problemlösung, die Beschreibung der Technologien "unbekannte Außenwelt". Ist die Domäne für die exemplarische Schaltung geschlossen zu beschreiben, d.h. im Sinne der geforderten Tiefe der Diagnosefindung eine hinreichende Symptom-Diagnose-Zuordnung aufzustellen, genügt der heuristische Ansatz den allgemeinen Anforderungen an die Leistungsfähigkeit eines wissensbasierten Systems, d.h. die Kenntnis bestimmter Bauteiltypen ist nicht erforderlich für die Unterscheidung "funktionsfähig - defekt".

Entscheidend wird die Einbeziehung allgemeinen Wissens allerdings für den Fall, daß das "Schaltungsdiagnose-Expertensystem" in Richtung generelle Platinendiagnose verallgemeinert werden soll, da für beliebige Schaltungen eine Symptom-Diagnose-Zuordnung sicherlich nicht in der für einen rein heuristischen Ansatz notwendigen Vollständigkeit aufgestellt werden kann. Die Aufgabe, kombinatorische Explosionen zu vermeiden, die auf die Einbeziehung allgemeineren Wissens in den Systementwurf zurückzuführen sind, stellt deshalb ein entscheidendes Problem für die Wissensrepräsentation (und folgende Realisierung) dar. Da

8) Grundlage ist die Beschreibung auf der Basis einer Symptom-Diagnose-Zuordnung.

die regelbasierte Lösungsfindung durch die Expansion des Suchraums an Grenzen zu stoßen scheint, bietet sich die Einbindung modellorientierter Repräsentationsverfahren als mögliche Alternative an. Die grundsätzlichen Prinzipien möglicher Ansätze modellgestützter Repräsentationsverfahren werden im folgenden Abschnitt beschrieben.

3.3 Ansätze für die Einbeziehung von Modellen in die Wissensdarstellung und -verarbeitung

Neben der Vermeidung kombinatorischer Explosionen, die, wie im Beispiel der Schaltungsdiagnose des vorigen Abschnitts, auf erhebliche Erweiterungen des Regelwerks durch Berücksichtigung allgemeinerer Regeln zurückzuführen sind, ist die Verarbeitung dynamischer Vorgänge ein wesentlicher Motivationsgrund für die Einbeziehung von modellbasierten Wissensbeschreibungen in den Entwurf eines Expertensystems.

Bei den klassischen Anwendungen wissensbasierter Technik wird angenommen, daß das betrachtete Problemfeld sich während der Bearbeitung nicht verändert, so wird z.B. bei Diagnoseapplikationen davon ausgegangen, daß der jeweilige Prüfling (oder bei medizinischen Problemen der Patient) im Verlauf der Konsultation seine Eigenschaften beibehält. Mit dieser statischen Betrachtung des Diagnosekandidaten ist der Problemraum der Lösung eingegrenzt, und kombinatorische Explosionen in der Lösungsfindung - zumindest solche, die auf Änderungen der physikalischen Parameter des betrachteten Systems zurückzuführen sind - ausgeschlossen.

Sind allerdings dynamische Vorgänge zu verarbeiten, ist das behandelte Problem von anderer Qualität, da ein statisches Regelwerk nicht zur vollständigen Interpretation des Zusammenhangs zwischen Ursachen und Wirkungen geeignet sein kann. Zur Begrenzung des Suchraums eignen sich in diesem Fall Wissensbeschreibungen, die numerisch basierte Modelle in den Repräsentationsformalismus einbeziehen [SHA87].

Die Aufgabe der Verarbeitung dynamischer Vorgänge stellt sich im wesentlichen so dar, daß aus einer begrenzten Menge verfügbarer Meßdaten eine vollständige Beschreibung des Gesamtzustands des behandelten Problems zu generieren ist. Algorithmi-

sche Verfahren erfordern eine hinreichend genaue mathematische Modellierung des betrachteten Systems, um eine, die Wirklichkeit möglichst genau wiedergebende, Lösung der Zustandsschätzung aus den verfügbaren Meßdaten zu erzielen. Für wissensbasierte Ansätze gilt im Gegensatz dazu, daß die Modellierung des Gesamtprozesses vernachlässigt werden kann, und, aus dem modellierbaren Wissen über Teilprozesse, eine begrenzte, z.B. numerisch basierte Beschreibung der gültigen Zusammenhänge aufgestellt wird. Durch diese Kombination heuristischen Wissens mit sogenanntem Deep-Level Knowledge ist dabei eine Verbesserung des Systemverhaltens gegenüber den klassischen Ansätzen zu erreichen [VEN86].

Die Kenntnis kausaler Beziehungen, z.B. von Naturgesetzen, ist eine erste zu berücksichtigende Stufe in der Repräsentation physikalischer Parameter. Die Darstellung derartigen Wissens kann dabei noch in Form von Regeln erfolgen, z.B. können Fallgesetz oder Impulserhaltungssatz verbal formuliert werden, und durch die Anpassung an den gewählten Repräsentationsformalismus ins Regelwerk einbezogen werden [MIL87]. Andererseits kann die Berücksichtigung solcher "Verhaltensweisen" aber auch durch mathematische Beziehungen erfolgen.

Häufig sind die betrachteten Vorgänge, bzw. die physikalischen Gesetzmäßigkeiten von Teilprozessen, mathematisch nicht geschlossen zu beschreiben⁹⁾, bzw. eine Modellierung durch Gleichungen nicht bekannt. Aus dem Wissen über den Prozeß lassen sich aber Proportionalitäten des Verhaltens der physikalischen Größen, sofern nicht unmittelbar bekannt, ableiten, die zur Aufstellung einfacher Beziehungen genutzt werden können.

Die Berücksichtigung solcher Proportionalitäten ist zum einen durch stationäre Gleichungen möglich, die, unter Verwendung der meßbaren Größen des Prozesses, sowie Randwerte und Nebenbedingungen repräsentieren Prozeßmerkmalen, die aus dem Erfahrungswissen des Experten zu gewinnen sind, entwickelt, Indikatoren zur Charakterisierung des Prozeßverhaltens liefern, ohne eine exakte Quantisierung einzubeziehen. In Abschnitt 4.3.2.3 wird ein derartiger Ansatz eingeführt, um die Risikowahrscheinlichkeit für das Eintreten eines gefährlichen Reaktionszustandes einer chemischen Umsetzung aus den Proportionalitäten zu Temperatur und relativer Reaktionsdauer zu bestimmen.

9) Diese Charakteristik bestimmt die Motivation, Expertensysteme für die geforderte Problemlösung einzusetzen.

Eine andere Form der Beschreibung verhaltensmäßiger Zusammenhänge ist durch die qualitative Modellbildung [KUI87] gegeben. Dabei werden die wechselseitigen Beziehungen zwischen den Prozeßgrößen in sogenannten Konfluenz-Gleichungen beschrieben, die lediglich die "Änderungsrichtung" z.B. einer Größe B, die sich aus der Veränderung einer Größe A ergibt, berücksichtigt, ohne die absoluten Quantitäten der Prozeßgrößen zu betrachten. Aus diesen Beziehungen sind die gültigen, d.h. "korrektes" Verhalten beschreibenden Fälle abzuleiten, auf denen dann die Interpretation des Prozeßverhaltens basiert. Die Regeln der qualitativen Addition sind in Tabelle 3.1 dargestellt.

$[dy]$ \ $[dx]$	+	0	-
+	+	+	-
0	+	0	-
-	-	-	+

Tabelle 3.1 : Regeln der qualitativen Addition

Ein Beispiel für die qualitative Modellierung von Prozessen ist die Betrachtung der Wechselwirkungen zwischen Drücken und Durchflüssen in einem Bypass-Strömungsrohr (Bild 3.4, [OYE87]), die durch 6 Konfluenz-Gleichungen beschrieben werden können. So gilt z.B. für die Beziehung zwischen Durchfluß F_1 und Druck P_1 am vorderen Verzweigungspunkt die Gleichung

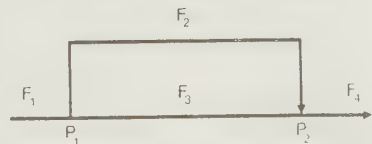


Bild 3.4 :
Bypass-Strömungsrohr

$$[dF_1] + [dP_1] = 0, \quad (3-1)$$

die so zu lesen ist, daß eine Minderung des Durchflusses eine Erhöhung des Druckes zur Folge hat, wobei die absoluten Werte dieser Größen nicht betrachtet werden, sondern nur der "Trend" berücksichtigt wird¹⁰⁾. Verhalten sich die Größen nicht gemäß dieser Beziehung, läßt sich daraus z.B. auf eine partielle Verstopfung des Strömungsrohrs schliessen, die durch die Modifizierung von Gl. (3-1) zu

10) Plausibel ist diese Deutung auch durch das Fehlen dimensionsregulierender Faktoren in der angegebenen Gleichung.

$$[dF_1] + [dP_1] + [dB] = 0 \quad \text{und} \quad (3-2)$$

$$[dB] = (+) \quad (3-3)$$

berücksichtigt werden kann. Ferner gilt für die Verhältnisse im Strömungsrohr z.B. die Beziehung

$$[dF_1] = [dF_2] + [dF_3] , \quad (3-4)$$

die so zu interpretieren ist, daß eine Zunahme des Flusses F_1 (im Normalfall) auch eine Zunahme der Flüsse F_2 und F_3 nach sich zieht.

Auf diese Weise lassen sich - im Idealfall - gesamte Prozesse relativ einfach beschreiben, ohne eine exakte mathematische Modellbildung zu erfordern. Eine Darstellung der behavioristischen Modellierung eines Rührkesselreaktors ist ebenfalls in [OYE87] zu finden. Ein ähnlicher Ansatz für diese Vorgehensweise bei der Fehlerdiagnose ist in [SHI85] beschrieben.

Die Einbeziehung exakter mathematischer Modelle in die Wissensrepräsentation [DAV82] wird insbesondere dann relevant, wenn Expertensysteme in eine umfassendere Programmumgebung eingebettet werden sollen, und die Anforderung besteht, daß das wissensbasierte Modul mit konventionellen Programmteilen "kommuniziert", d.h. nicht unmittelbar zu erfassende Größen als Eingangsdaten in ihrer Quantität vom Expertensystem benötigt werden, bzw. umgekehrt das Expertensystem dem konventionell implementierten Subprogramm Daten zur Verarbeitung "liefern" soll. Ein denkbarer Anwendungsbereich für eine solche Kopplung ist z.B. eine mathematische Modellbildung, die nur in der engeren Umgebung eines Arbeitspunktes verifizierbare Gültigkeit hat. Die Einbindung wissensbasierter Techniken in ein kombiniertes Gesamtsystem kann dabei zur Behandlung von "Randsituationen", bzw. von durch die Modellbildung nicht zu erfassenden Effekten, genutzt werden.

Die hier in konziser Darstellung aufgeführten Ansätze der Integration von Modellen in die Wissensrepräsentation, soll lediglich eine Übersicht über dieses Problemfeld in Form eines Ausblicks geben. Es ist zu erwarten, daß der Arbeitsbereich modellbasierte Expertensysteme, gerade im Zuge zunehmender Realzeit-Anwendungen wissensbasierter Systeme, in den kommenden Jahren einen wesentlichen Bestandteil der Forschung dieses KI-Schwerpunkts darstellen wird.

4. Erkennung kritischer Prozeßzustände chemischer Reaktionen

Eine wesentliche Aufgabe in der Führung und Überwachung technischer Systeme ist die Erkennung solcher Prozeßzustände, die für den Betrieb der Anlage gefährliche Situationen darstellen. Dabei ist es erforderlich, kritische Zustände derart frühzeitig zu erkennen, daß innerhalb einer verbleibenden Zeitspanne Maßnahmen zur Sicherung der Anlage durchzuführen sind. Ein zweites Kriterium, das diese Forderung beeinflußt, ist dabei die optimale Ausbeute bezüglich Menge und Qualität des Produktes, da die Erkennung anormaler Prozeßzustände in einem möglichst frühen Stadium diesbezüglich die Kontinuität einer optimierten Prozeßführung ermöglichen kann.

Die Führung chemischer Reaktoren stellt einen Anwendungsbereich der Prozeßautomatisierung dar [HIM78], in dem sich der Einsatz von Überwachungssystemen mit der Fähigkeit der Erkennung der Tendenz zu kritischen Zuständen empfiehlt, da die betrachteten Reaktionen häufig unter extremen Bedingungen stattfinden. Hohe Temperaturen, hoher Druck und katalytische Umsetzung sind dabei die charakteristischen Merkmale der Prozeßführung. Für eine wissensbasierte Lösung der Überwachungsaufgabe ziehen diese Voraussetzungen die Forderung nach sich, in angemessener Zeit auf Änderungen der Betriebszustände zu reagieren [FRÜ86], da exotherme Reaktionen unter diesen Bedingungen - die Abhängigkeiten zwischen den Prozeßgrößen sind durch starke Nichtlinearitäten charakterisiert - eine hohe parametrische Empfindlichkeit aufweisen. So können z.B. bereits geringe Abweichungen der Betriebstemperatur unerwünschte Temperaturspitzen verursachen, die z.B. zur vorzeitigen Deaktivierung des Katalysators beitragen. Ferner kann das Auftreten von parasitären Nebenreaktionen die Folge solcher Abweichungen sein. Der Anstieg der Temperatur kann außerdem zu einem Druckanstieg im Reaktionsgefäß führen, der im Extremfall das Bersten des Behälters nach sich zieht.

[SCH82] behandelt am Beispiel von Chargenprozessen, die sich durch eine Formalkinetik 1. Ordnung beschreiben lassen, die frühzeitige Erkennung kritischer Reaktionszustände mit Hilfe modellgestützter Meßverfahren, die aus den zur Verfügung stehenden unvollständigen Meßgrößen den Gesamtzustand der betrachteten Anlage rekonstruieren. Die Aufgabe der Erkennung gefährlicher Zustände wird dabei als Gegenstand der Mustererkennung betrachtet. Die angeführten Methoden werden, exemplarisch für konventionelle Überwachungsverfahren, im folgenden zusammen-

gefaßt dargestellt. Dabei wird von der allgemeinen Struktur der Prozeßüberwachungsaufgabe (Bild 4.1) ausgegangen, wobei die Alarmvariable α , die in Form eines Funktional der Meßwerte

$$\alpha = A\{z\} \quad (4-1)$$

gegeben ist, sich auf den diskreten Wertebereich

$$\alpha = \begin{cases} 1 & \text{für gefährliche Zustände} \\ 0 & \text{für unkritische Zustände} \end{cases} \quad (4-2)$$

beschränkt. Nimmt die Alarmvariable α im Verlauf einer Reaktion den Wert 1 an, so werden aufgrund des Vorliegens eines kritischen Prozeßzustandes Maßnahmen ergriffen.



Bild 4.1 : Allgemeine Struktur der Überwachungsaufgabe

Die Anwendung konventioneller, algorithmenbasierter Methoden zur Erkennung kritischer Zustände erfordert die genaue Kenntnis des Modells des betrachteten Prozesses. Für viele chemische Reaktionen existieren jedoch keine oder nur unzureichend genaue mathematische Modelle, obwohl sie großtechnisch gefahren werden. Gegebene Voraussetzungen sind dabei Teilmodelle des Gesamtprozesses, die z.B. auf Naturgesetzmäßigkeiten basieren, sowie Erfahrungswerte aus dem Betrieb der Anlage, die sich z.B. auf Messungen während des Ablaufs der Reaktion stützen. Da ferner die Überwachungsaufgabe selbst, sowie die Methoden der Mustererkennung Ähnlichkeit zu Diagnoseproblemen aufweisen, bietet der Einsatz von Expertensystemtechniken [PEN87] die Möglichkeit, orientiert an den genannten gegebenen Voraussetzungen, eine wissensbasierte Lösung der Aufgabe zu erzielen [AHR86].

4.1 Exemplarisch betrachtete Prozesse : Diskontinuierlich betriebene Rührkesselreaktoren

Zahlreiche chemische Prozesse werden im Chargenbetrieb durchgeführt. Dabei läßt sich, bezogen auf die Aufgabe der Erkennung kritischer Zustände, zwischen zwei charakteristischen Fällen des Auftretens gefährlicher Reaktionsverläufe unterscheiden.

Chemische Umsetzungen, die eine starke Wärmetönung, d.h. hohe freiwerdende Energie, aufweisen, sind dadurch gekennzeichnet, daß mit fortschreitendem Reaktionsverlauf die Temperatur ansteigt. Die Selbstbeschleunigung des Reaktionsverlaufs kann dabei unter bestimmten Betriebsbedingungen zu explosionsartigem Temperaturanstieg führen. Ein Beispiel dafür ist eine Polymerisationsreaktion, die unterschiedliche Temperaturverläufe in Abhängigkeit von einer Initiatorkonzentration $[I]_0$ aufweist (Bild 4.2).

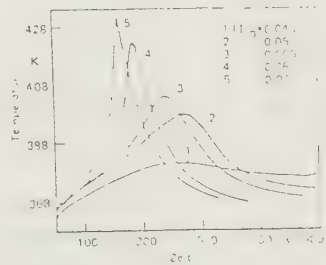


Bild 4.2 : Durchgehen einer Polymerisationsreaktion in Abhängigkeit von einer Initiatorkonzentration

Eine andere Klasse chemischer Umsetzungen weist die Charakteristik auf, daß die Reaktionsführung in bestimmten Temperaturbereichen durchgeführt werden muß, da eine Abweichung der Betriebstemperatur von der Solltemperatur zur Zersetzung des Produktes durch Anspringen einer unerwünschten Folgereaktion führen kann. Die Sulfonierung ist ein Beispiel für diese Klasse von Reaktionen, da, in Abhängigkeit von der Einfülltemperatur des Edukts, die Reaktion Temperaturbereiche erreichen kann, in der die Zersetzung des Produktes beginnt (Bild 4.3).

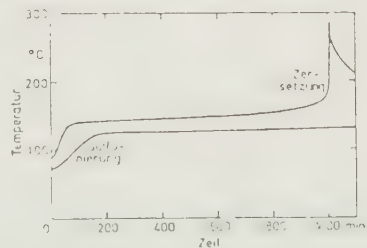
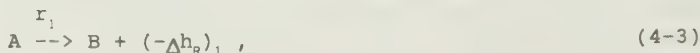


Bild 4.3 : Verlauf einer Sulfonierungsreaktion mit dem Anspringen einer Folgereaktion bei erhöhter Einfülltemperatur

Die mathematische Modellbildung derartiger Batch-Prozesse erfolgt über die Bilanzen von Stoffmenge und Energie. Für die Synthesereaktion nach der erwähnten Formalkinetik 1. Ordnung



lauten die Energie- und Stoffmengenbilanz des Reaktors

$$C \cdot (dT/dt) = k_w \cdot F_K \cdot (T_K - T) + (-\Delta h_R)_1 \cdot r_1(T, c_A) \cdot V, \quad T(0) = T_0, \\ (dc_A/dt) = -r_1(T, c_A), \quad c_A(0) = c_{A0}, \quad (4-4)$$

mit der Reaktionsgeschwindigkeit

$$r_1(T, c_A) = k_{10} \cdot e^{-E_1/R \cdot T} \cdot c_A, \quad (4-5)$$

gemäß dem bekannten Arrhenius-Ansatz. Dabei bezeichnet C eine zusammenfassende Konstante der auftretenden Wärmekapazitäten, k_w eine Wärmedurchgangszahl, F_K die Kühlfläche, die Differenz $(T_K - T)$ die Temperaturdifferenz zwischen Kühlfläche und Reaktor, V das Reaktorvolumen und $(-\Delta h_R)_1$ die Reaktionsenthalpie. T und c_A bezeichnen Temperatur und Produktmenge der Reaktion¹⁾.

Die Normierung dieses Modells einer exothermen Reaktion durch Einführung der dimensionslosen Variablen

$$x_1 = (T - T_K) / T_K \cdot E_1 / R \cdot T_K, \quad (4-6)$$

$$x_2 = (c_{A0} - c_A) / c_{A0} \quad \text{und} \quad (4-7)$$

$$\tau = k_{10} \cdot e^{-E_1/R \cdot T_K} \cdot t \quad (4-8)$$

als dimensionslose Zeit, führt auf eine Form nichtlinearer Zustandsdifferentialgleichungen

1) In dieser Modellbildung sind einige vereinfachende Annahmen berücksichtigt.

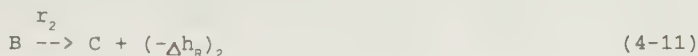
$$(dx_1/d\tau) = a \cdot r_1(x_1, x_2) - b \cdot x_1, \quad x_1(0) = x_{10},$$

$$(dx_2/d\tau) = r_1(x_1, x_2), \quad x_2(0) = 0, \quad (4-9)$$

$$r_1(x_1, x_2) = (1 - x_2) \cdot e^{\frac{x_1}{(1 + x_1/c)}}, \quad (4-10)$$

mit denen die Reaktion gemäß Bild 4.2 in geeigneten Bereichen der Parameter a, b, c, x_{10} zu beschreiben ist.

Wird ferner berücksichtigt, daß neben der Synthesereaktion eine Zersetzungsreaktion gemäß



stattfindet, führt die Einführung einer weiteren Zustandsvariablen x_3 auf ein erweitertes Modell der Form

$$(dx_1/d\tau) = -b \cdot x_1 + a_1 \cdot r_1 + a_2 \cdot r_2, \quad x_1(0) = x_{10},$$

$$(dx_2/d\tau) = r_1, \quad x_2(0) = 0,$$

$$(dx_3/d\tau) = r_2, \quad x_3(0) = 0, \quad (4-12)$$

$$r_1 = (1 - x_2) \cdot e^{\frac{x_1}{(1 + x_1/c)}},$$

$$r_2 = d \cdot (x_2 - x_3) \cdot e^{\frac{c_2 \cdot x_1 / c_1 \cdot (1 + x_1/c)}{}} \quad (4-13)$$

mit dem Reaktionsverläufe gemäß Bild 4.3 zu beschreiben sind.

Bereits mit derartigen einfachen Modellen können die zeitlichen Verläufe der Zustandsgrößen der exemplarisch angeführten Reaktionen im geeigneten Wertebereich der Parameter richtig wiedergegeben werden. Dabei muß allerdings berücksichtigt werden, daß die Modelle (4-9) und (4-12), aufgrund einiger Vereinfachungen in der Modellbildung, nur in der Umgebung eines bestimmten Arbeitspunktes verifizierbare Ergebnisse liefern werden.

4.2 Die Anwendung konventioneller Methoden zur Lösung der Überwachungsaufgabe : Erkennung kritischer Zustände mit Methoden der Mustererkennung auf der Basis modellgestützter Meßverfahren

Das in Bild 4.1 dargestellte allgemeine Überwachungssystem zur Generierung einer Alarmvariablen α ist unter Anwendung der Mustererkennung charakterisiert durch einen zweistufigen Aufbau, bestehend aus Extraktion und Klassifikation (Bild 4.4). Die Extraktionsstufe dient dabei der Verarbeitung der Meßgrößen, die als Muster aufgefaßt werden, und der Generierung eines Schätzwertes, der die Charakterisierung des momentanen Prozeßzustandes ermöglicht. Die Entscheidungsstufe dient der Zuordnung dieses Schätzwertes zu einer Musterklasse, die angibt, ob sich der Prozeß in unkritischem oder gefährlichem Zustand (Alarmvariable α) befindet.

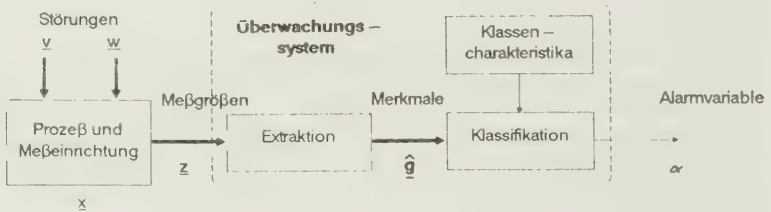


Bild 4.4 : Prozeßüberwachungssystem auf der Basis der Mustererkennung

Wird für die Modelle (4-9) und (4-12) der normierten Bilanzengleichungen berücksichtigt, daß Einflüsse nicht betrachteter Effekte, wie Ungenauigkeiten der Parameter und unvorhergesehene Störungen, dazu führen, daß das Verhalten im Laufe der Zeit vom tatsächlichen Verlauf der Reaktion abweichen kann, so führt die Berücksichtigung dieser Störeinflüsse durch stochastische Eingangsgrößen auf das verallgemeinerte Modell der Bilanzengleichungen

$$(d\mathbf{x}/d\tau) = \mathbf{f}(\mathbf{x}, \mathbf{u}) + \mathbf{G}\mathbf{v} , \quad \mathbf{x}(0) = \mathbf{x}_0 , \quad (4-14)$$

wobei die zufälligen Einflüsse, die als mittelwertfreier Gaußscher Prozeß mit abschätzbarer Kovarianzmatrix aufgefaßt werden, im Vektor $\mathbf{v}$ zusammengefaßt sind, der über die Störmatrix $\mathbf{G}$ bewertet wird.

Die meßbaren Größen des Prozesses werden in der allgemeinen Form durch die Beziehung

$$\underline{z}(k) = \underline{h}(\underline{x}(k)) + \underline{w}(k) \quad (4-15)$$

berücksichtigt, wobei die Verarbeitung der Meßdaten in den äquidistanten Abtastzeitpunkten τ_k mit einem Prozeßrechner vorausgesetzt ist. Der Vektor $\underline{w}(k)$ enthält die den tatsächlichen Meßgrößen überlagerten Störanteile, von denen wiederum einschränkend angenommen wird, daß diese einen mittelwertfreien Gaußschen Zufallsprozeß mit abschätzbarer Kovarianzmatrix bilden.

Für die meisten Chargenprozesse kann davon ausgegangen werden, daß nur die Reaktionstemperatur als meßbare Größe zur Verfügung steht. Damit degeneriert die allgemeine Meßgleichung (4-15) zu

$$z = x_1 + w, \quad (4-16)$$

wobei x_1 die dimensionslose Temperatur der Beispielreaktionen (4-9) und (4-12) darstellt, und durch w ein Störanteil berücksichtigt wird.

Für die Störterme $\underline{v}$ und w in Zustands- und Meßgleichung wird vorausgesetzt, daß sie nicht miteinander korreliert sind. Die Grundlage für die Vorausberechnung der Reaktionsverläufe im Batch-Reaktor bildet nunmehr der Erwartungswert der nichtlinearen Zustandsdifferentialgleichung, der, wenn die Mittelwertfreiheit des Störprozesses $\underline{v}$ berücksichtigt wird, bezüglich der allgemeinen Form (4-14) lautet

$$(d\hat{\underline{x}}/d\tau) = \underline{f}(\hat{\underline{x}}, \underline{u}), \quad \hat{\underline{x}}(0) = \hat{\underline{x}}_0. \quad (4-17)$$

Aufgrund des Einwirkens der Störgröße $\underline{v}$ besitzt auch der Zustand $\underline{x}$ statistische Eigenschaften, die näherungsweise als normalverteilter Zufallsprozeß angenommen werden können. Die Abschätzung des Schätzfehlers kann mit der Kovarianzmatrix

$$\underline{P} = E\{[\underline{x} - \hat{\underline{x}}][\underline{x} - \hat{\underline{x}}]^T\} \quad (4-18)$$

erfolgen.

Die in diese Modellbildung eingehenden Annahmen bilden die Basis, um mittels modellgestützter Meßverfahren aus den unvollständigen Meßgrößen den Gesamtzustand der betrachteten Anlage zu rekonstruieren und den Bereich kritischer Zustände abzugrenzen.

Für die betrachtete exotherme Reaktion gemäß Modell (4-9) wird vorausgesetzt, daß die Meßgleichung (4-16) gilt, d.h. nur die Temperatur x_1 steht als Meßgröße zur Verfügung. Ein geeignetes Kriterium für einen gefährlichen Reaktionsverlauf ist dann das beschleunigte Ansteigen der Temperatur, d.h. durch

$$(dx_1/d\tau) > 0 \quad \text{und} \quad (d^2x_1/d\tau^2) > 0 \quad (4-19)$$

wird das Gebiet kritischer Zustände definiert, was zu einem Überwachungssystem einfachster Struktur führt, da, bei numerischer Differentiation über geeignete Meßwertfolgen $z(k)$, kein Prozeßmodell benötigt wird [HUB82].

Um den Hochpaßcharakter der zweimaligen Differentiation auszuschalten, wird nicht der Meßwert selbst verarbeitet, sondern durch seine Auswertung zunächst der gesamte Prozeßzustand $\underline{x}$ rekonstruiert, wobei ein als Ein-Schritt-Prädiktor ausgelegtes erweitertes Kalman-Filter²⁾ verwendet wird, um der nichtlinearen Struktur der Modelle (4-9) und (4-12) zu genügen.

Mit den Schätzwerten des Zustandsvektors lassen sich dann die Komponenten des Merkmalvektors

$$\begin{aligned} \hat{g}_1 &= (d\hat{x}_1/d\tau) \\ \hat{g}_2 &= (d^2\hat{x}_1/d\tau^2) \end{aligned} \quad (4-20)$$

näherungsweise direkt aus den Modellgleichungen (4-9) berechnen und somit die expliziten Differentiationen vermeiden.

Der Wert der ersten Ableitung der Temperatur ist explizit durch die rechte Seite der Temperaturgleichung des Modells (4-9) gegeben. Einsetzen der Schätzwerte des Filters ergibt

$$\hat{g}_1(\hat{x}_1, \hat{x}_2) = a \cdot \hat{r}_1(\hat{x}_1, \hat{x}_2) - b \cdot \hat{x}_1. \quad (4-21)$$

2) Die Gleichungen des erweiterten Kalman-Filters sind in Anhang A dokumentiert.

Der Wert der zweiten Ableitung ergibt sich durch Anwendung der Kettenregel auf die rechte Seite der Temperaturgleichung und Einsetzen der Zustandsschätzwerte in die Lösung zu

$$\hat{g}_2(\hat{x}_1, \hat{x}_2) = (a^2/(1 + \hat{x}_1/c)^2 \cdot [\hat{r}_1^2(\hat{x}_1, \hat{x}_2) - m_1(\hat{x}_1) \cdot \hat{r}_1(\hat{x}_1, \hat{x}_2) + m_2(\hat{x}_1)] , \quad (4-22)$$

wobei m_1 und m_2 gegeben sind durch

$$\begin{aligned} m_1(\hat{x}_1) &= (b/a) \cdot [\hat{x}_1 + (1 + \hat{x}_1/c)^2 \cdot \hat{x}_1/(1 + \hat{x}_1/c) \cdot (1 + (1/b) \cdot e)] , \\ m_2(\hat{x}_1) &= (b/a)^2 \cdot (1 + \hat{x}_1/c)^2 \cdot \hat{x}_1 . \end{aligned} \quad (4-23)$$

Es ist ferner zu berücksichtigen, daß mittels

$$\hat{r}_1(\hat{x}_1, \hat{x}_2) = (1 - \hat{x}_2) \cdot e^{\hat{x}_1/(1 + \hat{x}_1/c)} \quad (4-24)$$

die Schätzwerte des erweiterten Kalman-Filters auch in die Beziehung zur näherungsweisen Bestimmung der Reaktionsgeschwindigkeit eingehen.

Die Klassifikation des Merkmalvektors (4-20) erfolgt auf der Grundlage des einfachen Kriteriums (4-19), das als geometrische Entscheidungsregel interpretiert werden kann. Dazu ist allerdings eine hinreichend genaue Berechnung des Merkmalvektors aus den Meßgrößen erforderlich.

Kann jedoch nicht von der exakten Kenntnis des Merkmalvektors ausgegangen werden, was unter echten Prozeßbedingungen realistisch sein wird, so ist das Entscheidungsrisiko, das mit dem Auftreten von Unsicherheiten verbunden ist, zu minimieren. In diesem Fall wird aufgrund einer statistischen Analyse die Wahrscheinlichkeit berechnet, daß sich der Prozeßzustand in einem unkritischen Bereich befindet, und zur Bestimmung des Risikos einer Entscheidung verwendet, wobei die möglichen Kosten einer Fehlentscheidung berücksichtigt werden³⁾. Dieser aufwendigere Ansatz berücksichtigt das gesamte a-priori-Wissen über den Prozeß und zeigt daher in der Regel auch das bessere Verhalten gegenüber der deterministischen Klassifikation⁴⁾.

3) Es sollen sowohl zu späte oder nicht erfolgenden Notabschaltungen vermieden werden, als auch unnötige Fehlabschaltungen im Normalbetrieb.

4) Die verwendeten Gleichungen für die statistisch basierten Klassifikationskriterien sind in Anhang B zusammengefaßt.

Das Eintreten parasitärer exothermer Folgereaktionen gemäß Modell (4-12) läßt sich erkennen, wenn parallel geschaltete Filter die Zustandsschätzwerte berechnen, wobei ein Filter den bestimmungsgemäßen Normalbetrieb berücksichtigt. Für jeden möglichen Störfall ist ein weiteres Filter zu verwenden, daß ein Modell dieses Störfalls enthält. Aufgrund einer statistischen Analyse wird dann die Wahrscheinlichkeit berechnet, ob sich die Zustände in einem unkritischen Gebiet befinden, und die Klassifikation auf der Basis minimalen Risikos durchgeführt⁴⁾.

Die [SCH84] entnommenen und hier zusammenfassend beschriebenen Methoden zur Erkennung kritischer Reaktionszustände in chemischen Reaktoren zeigen bei der dort vorgenommenen Überprüfung durch Simulationen geeignetes Verhalten zur Lösung der gestellten Aufgabe. Die (Vor- und) Nachteile als Konsequenzen der Anwendung dieser Überwachungssysteme sollen an dieser Stelle nicht im Detail diskutiert werden, als wesentlich erscheint jedoch die mit dem statistischen Ansatz verbundene aufwendige Realisierung, sowie die Problematik des Signalverhaltens der vereinfachten Verfahren.

4.3 Erkennung kritischer Reaktionszustände durch Anwendung wissensbasierter Systeme

Im folgenden werden, aufbauend auf der Begründung für einen wissensbasierten Systementwurf, zwei mögliche Verfahren zur Lösung der Prozeßüberwachungsaufgabe mit Hilfe von Expertensystemen vorgestellt, wobei vorausgesetzt wird, daß für die betrachteten Reaktionen keine mathematischen Modelle existieren. Der erste Systementwurf basiert dabei auf einem rein heuristischen Ansatz; im zweite Entwurf wird ein Tendenzmodell in Form einer einfachen stationären Gleichung eingeführt, dessen Parameter aus dem Wissen über den betrachteten Prozeß ausgelegt werden. Einige Betrachtungen zur notwendigen Wissensakquisition und -umsetzung schliessen den Abschnitt ab.

4.3.1 Motivation für den Einsatz von Expertensystemen

Algorithmenorientierte Verfahren zur Erkennung gefährlicher Zustände in chemischen Reaktoren erfordern die genaue Kenntnis des Modells des betrachteten Reaktionsprozesses hinsichtlich der Struktur, d.h. beispielsweise die Ordnung des Prozesses und der Grad der auftretenden Nichtlinearitäten sollten zumindest bekannt sein. Unvollständige Kenntnis der Systemparameter ist durch die Verwendung geeigneter Parameterschätzverfahren zu kompensieren. Häufig existieren für chemischen Reaktionen, obwohl sie apparatetechnisch durchgeführt werden, jedoch nur unzureichende, bzw. in begrenzten Bereichen gültige, oder keine mathematischen Modelle. Die Führung der Anlage basiert dann im wesentlichen auf den Erfahrungswerten des Betriebspersonals, die sich aus der Interpretation von Messungen während des Reaktionsverlaufs und dem Wissen über bestimmte Situationen, sowie der Kenntnis kausaler Zusammenhänge, wie z.B. Naturgesetzen, zusammensetzen. Mit dem Wissen der am Betrieb der Anlage beteiligten Wartenfahrer, Ingenieure und Chemiker, steht somit eine Grundlage zur Verfügung, die Aufgabe der Erkennung kritischer Reaktionszustände mit einem Expertensystem-Ansatz zu lösen.

Für die in Abschnitt 4.1 exemplarisch betrachteten Reaktionen, mit denen die Anwendung konventioneller Methoden demonstriert wird, stehen Modelle zur Verfügung (Gln. (4-9) u. (4-12)), die die Reaktionsverläufe hinreichend genau beschreiben. Damit ist für diese speziellen Reaktionen eine algorithmische Lösung der gestellten Aufgabe möglich, die, trotz des evtl. hohen Realisierungstechnischen Aufwands, der wissensbasierten Problemlösung vorgezogen werden sollte. Trotzdem lassen sich aus den dort durchgeführten Betrachtungen Aspekte extrahieren, die bei der Verallgemeinerung zu beliebigen Reaktionen - ohne (vorhandene) Modellierung - wichtige Ansatzpunkte für die Rechtfertigung des wissensbasierten Problemlösungsansatzes darstellen.

Betrachtet man die Definition eines gefährlichen Reaktionsverlaufs als beschleunigten Temperaturanstieg (Gl. (4-19)), so ist dazu zu bemerken, daß der absolute Wert der momentanen Betriebstemperatur durch dieses Kriterium nicht berücksichtigt wird. Unter der Voraussetzung, daß die Meßgröße Temperatur nur positive Werte annimmt, was praxisbezogen eine realistische Annahme ist, können Reaktionsbedingungen auftreten, bei denen das Kriterium zu einer falschen Klassifikation führt, was durch zwei einfache Beispiele verdeutlicht werden soll.

Steigt z.B. die Temperatur (näherungsweise) linear, dann befindet sich die Reaktion gemäß Kriterium (4-19) in einem Gebiet unkritischer Zustände. Die absolute Betriebstemperatur kann sich aber bereits in Wertebereichen befinden, in denen ein weiteres Ansteigen die Sicherheit der Anlage gefährden kann, oder zumindest, durch resultierenden Druckanstieg, z.B. die Öffnung von Sicherheitsventilen zur Folge hat. Ferner ist die "harmloseste" Folge, eine Beeinträchtigung der Produktausbeute bezüglich der Qualität, denkbar.

Verändert sich die Temperatur gemäß einer Normalparabel 2. Grades, so stellt jeder Zustand laut Kriterium (4-19) einen gefährlichen Reaktionszustand dar. Befindet sich die absolute Momentantemperatur noch in einem sicheren Bereich, so würde die Auslösung eines Alarms und nachfolgender Maßnahmen eine Beeinträchtigung der Produktausbeute bezüglich der Menge verursachen, im Extremfall sogar ein Löschen der Reaktion zur Folge haben können.

Angemerkt sei hier noch, daß exotherme Reaktionen die Charakteristik aufweisen, daß der Temperaturverlauf von der Zufuhr der Ausgangsstoffe abhängig ist. Mit abnehmender Stoffmenge der Edukte im Reaktor klingt die Heftigkeit der Reaktion "natürlich" ab, was das Auftreten der genannten Extremfälle zwar einschränkt, das Eintreten von Betriebsbedingungen, die z.B. mittelbar den Zustand eines Katalysators beeinflussen, aber nicht ausschließt. Auch solche Reaktionen, bei denen ein räumliches Temperaturprofil eingestellt werden soll (in der Regel in kontinuierlich betriebenen Durchfluß-Reaktoren), sind durch dieses Merkmal charakterisiert. Hier erweisen sich experimentell gefundene Methoden als geeignet, so z.B. die Vermeidung von Temperaturspitzen durch Katalysatorverdünnung bei der Methanisierungsreaktion [DAN85].

Durch die Berücksichtigung von Rand- und Nebenbedingungen in einem wissenschaftlichen Problemlösungsansatz, kann der Gültigkeitsbereich des Kriteriums (4-19) derart begrenzt werden, daß bestimmte, aufgrund von Erfahrungen bekannte Situationen, in denen das Auslösen eines "falschen" Alarms vermieden wird, oder umgekehrt die Auslösung eines Alarms während eines scheinbar "harmlosen" Reaktionsverlaufs initiiert wird, erfaßt werden.

Ist das Modell des betrachteten Prozesses bekannt, so stützt sich die Extraktion der Merkmale aus den Meßgrößen und die Klassifikation dieser Merkmale in Klassen von unkritischen und gefährlichen Zustandsbereichen auf die Gültigkeit dieses Modells. Beim statistischen Ansatz wird zusätzlich berücksichtigt, daß durch die statistischen Kennwerte der Störprozesse, hier insbesondere die Kovarianzmatrizen, das Vertrauen in die

Gültigkeit des Modells ausgedrückt wird. Die genaue Kenntnis der Kovarianzmatrizen der Störprozesse ist in der Regel nicht erreichbar⁵⁾, gewöhnlich erfolgt die Auslegung aufgrund von groben Abschätzungen und Erfahrungswerten.

In einem wissensbasierten Problemlösungsansatz, der kein Prozeßmodell berücksichtigt, kann durch die Vergabe von Konfidenzfaktoren, die sich auf das Erfahrungswissen des/der Experten stützen, das Vertrauen in die Gültigkeit von Aussagen und Beziehungen berücksichtigt werden. Damit steht auch ein vereinfachter Ansatz zur statistischen Auswertung zur Verfügung.

Das Eintreten parasitärer Neben- oder Folgereaktionen wird bei der Anwendung modellgestützter Verfahren durch die Verarbeitung der Meßwerte durch parallel geschaltete Filter, die Normalbetrieb und mögliche Störfälle wiedergeben, erfaßt. Voraussetzung für die Zuverlässigkeit dieses Lösungsansatzes ist allerdings die genaue Kenntnis der Charakteristika möglicher Störfälle. Für gewöhnlich ist das Wissen über den bestimmungsgemäßen Normalbetrieb zu erfassen, das Wissen über Störfälle aber nur unvollständig bekannt. Durch geeignete Bedingungen, die in einem wissensbasierten Problemlösungsansatz berücksichtigt werden, ist eine Begrenzung des zu behandelnden Problemraums derart zu erreichen, daß aufgrund der Merkmalsgenerierung eine Klassifikationsentscheidung ermöglicht wird.

Die in Abschnitt 4.2 als Fazit genannten Nachteile der Extremfälle konventioneller Verfahren, einerseits Realisierungsaufwand und andererseits ungünstiges Signalverhalten, lassen sich unter Anwendung eines wissensbasierten Ansatzes umgehen. Die Vermeidung numerisch ungeeigneter Methoden, wie die expliziten Differentiationen, durch auf Fakten gestützte Schlußfolgerung, ermöglicht eine direkte Generierung der Merkmale, sofern in ausreichendem Maße Rand- und Nebenbedingungen in den Inferenzprozeß einbezogen werden können. Die Klassifikation der Merkmale in Bereiche von unkritischen oder gefährlichen Zustandsgebieten aufgrund von wissensbasierten Kriterien, kann dazu dienen, den relativ hohen Realisierungsaufwand der Algorithmen zur statistischen Analyse⁶⁾ zu umgehen.

5) Die Kovarianzmatrix der Meßstörung kann allerdings auf Grundlage der Kennwerte des verwendeten Meßgerätes ausgelegt werden.

6) Siehe auch Anhang B.

Der wissensbasierte Problemlösungsansatz zur Erkennung kritischer Reaktionszustände in chemischen Reaktoren, unter der Voraussetzung unvollständiger Modellierung der betrachteten Prozesse, läßt erwarten, daß eine zufriedenstellende Lösung der Aufgabe unter Berücksichtigung der genannten Bedingungen erreichbar ist. Die Durchführung der Wissensgewinnung und -umsetzung ist aber an die Voraussetzung gebunden, daß ein bestimmter chemischer Reaktionsprozeß behandelt wird, was impliziert, daß ein geeigneter Experte, möglichst zur ständigen Befragung, verfügbar ist. Da im Rahmen dieser Arbeit diese Voraussetzung nicht bestand, wird die Diskussion der folgenden relevanten Aspekte einerseits auf einem allgemeineren Niveau, andererseits in Bezugnahme auf die bei der konventionellen Lösung betrachteten Reaktionen⁷⁾, durchgeführt.

4.3.2 Struktur des wissensbasierten Prozeßüberwachungssystems

Die Realisierung von Expertensystemen für die Überwachung derartiger komplexer Prozesse wie chemische Umsetzungen, erfordert, daß der Wissensakquisition eine sorgfältige Vorbereitung in Form des Entwurfs einer Systemgrundstruktur vorangeht (vgl. Abschnitt 2.3.2). Ferner ist generell zu berücksichtigen, daß die Automatisierung komplexer Prozesse grundsätzlich eine methodische Vorgehensweise bei der Planung der Durchführung des Verfahrens verlangt [BR086]. Die Entwicklung der Systemstruktur für die ausgewählte Teilaufgabe der Überwachung soll im folgenden behandelt werden.

7) Es sei hier nochmals angemerkt, daß für Aufgaben, für die die Möglichkeit einer algorithmenbasierten Lösung besteht, diese der wissensbasierten Lösung vorzuziehen ist.

4.3.2.1 Problemstellung

Die Erkennung kritischer Prozeßzustände stellt eine typische diagnostische Realzeit-Aufgabe [FRÜ86] dar. Damit ergibt sich für den Entwurf eines wissensbasierten Überwachungssystems als erste Konsequenz, einem mehrstufigen Aufbau zu genügen, um zeitkritische Vorgänge hinreichend schnell erfassen und verarbeiten zu können. Dabei ist zu berücksichtigen, daß das gestellte Problem eine hybride Aufgabe darstellt, d.h. die Meßdatenverarbeitung und Merkmalserkennung von der Meßdatenerfassung separiert betrachtet werden kann, wobei letztere in konventioneller DV-Technik ausgeführt wird. Das Expertensystem zur Überwachung stellt aufgrund des in der Wissensbasis abgebildeten Expertenwissens dann Schlußfolgerungen über die aktuellen Meßdaten auf. Die allgemeine Grundstruktur eines wissensbasierten Überwachungssystems ist in Bild 4.5 dargestellt.

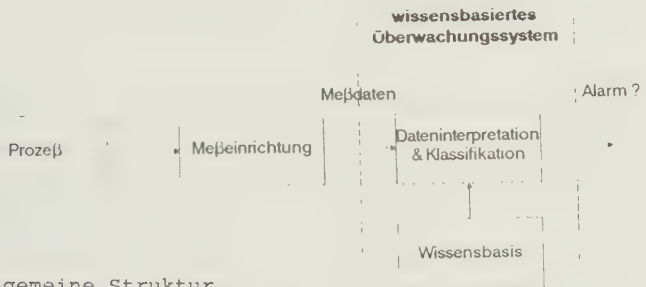


Bild 4.5 : Allgemeine Struktur wissensbasierter Überwachungssysteme

Betrachtet man den exemplarischen Temperaturverlauf gemäß Bild 4.2, so ist ein wesentliches Merkmal dadurch gegeben, daß in niedrigen Temperaturbereichen die Wahrscheinlichkeit für das Eintreten eines Störfalls nahe Null liegt, und erst mit steigender Temperatur zunimmt. In Bereichen abnehmender Temperatur nimmt das Gefahrenrisiko dann ebenfalls wieder ab. Ebenso ist der Temperaturverlauf gemäß Bild 4.3 dadurch gekennzeichnet, daß das Eintreten der Sekundärreaktion mit dem Erreichen ganz bestimmter Temperaturbereiche verbunden ist. Diesen Tatsachen kann dadurch Rechnung getragen werden, in dem der eigentlichen

Dateninterpretation und Merkmalsklassifikation eine Selektion der Meßgrößen in Gebiete geringerer und höherer Wahrscheinlichkeit des Eintretens einer gefährlichen Situation vorgeschaltet wird, die sich z.B. auf die absolute Momentantemperatur stützen kann.

Die nicht vorhersehbare Suchtiefe eines, z.B. regelinterpretierenden Inferenzmechanismus' hat zur Folge, daß bei schnell ablaufenden Reaktionen unter bestimmten Bedingungen Meßdaten nur unvollständig erfaßt werden können. Da die frühzeitige Erkennung der Merkmale eines gefährlichen Reaktionsverlaufs sich auf die möglichst vollständige Erfassung der Meßgrößen stützt, ist durch einen rein heuristischen Ansatz zur Merkmalsgenerierung unter Umständen die Aufgabe nicht zu lösen. In diesem Fall kann die Merkmalsextraktion auf der Basis eines einfachen Tendenzmodells, das aus dem Wissen über den Prozeß entwickelt wird, die rechtzeitige Erkennung einer kritischen Situation unterstützen.

Das "Verhalten" exothermer chemischer Umsetzungen ist gekennzeichnet durch die gegenseitige Beeinflussung der Prozeßparameter und der Abhängigkeit der Prozeßvariablen von den Parametern, die von hoher Empfindlichkeit sein kann. Wird eine Reaktion mathematisch modelliert, so zeigt sich ferner, daß bereits bei so einfachen Modellen, wie durch die Glgn. (4-9) und (4-12) beschrieben, Nichtlinearitäten hohen Grades den Prozeß charakterisieren. Steht, wie in den meisten praktischen Fällen, nur die Temperatur als Meßgröße zur Verfügung, so ist die Klassifikation des Prozeßverhaltens auf relativ geringe Information gestützt. Der wissenbasierte Ansatz zur Klassifikation scheint hier den wesentlichen Vorteil gegenüber konventionellen Methoden zu bergen, da Rand- und Nebenbedingungen explizit in die Entscheidung einbezogen werden können.

4.3.2.2 Struktur des wissensbasierten Überwachungssystems auf der Grundlage eines heuristischen Ansatzes

Bild 4.6 zeigt den dreistufigen Aufbau eines Expertensystems zur Erkennung gefährlicher Reaktionszustände auf der Basis eines rein wissenbasierten Entwurfs. Aus Effizienzgründen, um die Suchtiefe des Inferenzmechanismus' zu begrenzen, ist die Generierung geeigneter Merkmale in die Selektionsstufe zur Erzeugung eines Situationsmusters, und in die Interpretationsstufe zur Umsetzung des Musters in geeignete Klassenmerkmale

heuristisch basierte Systemstruktur

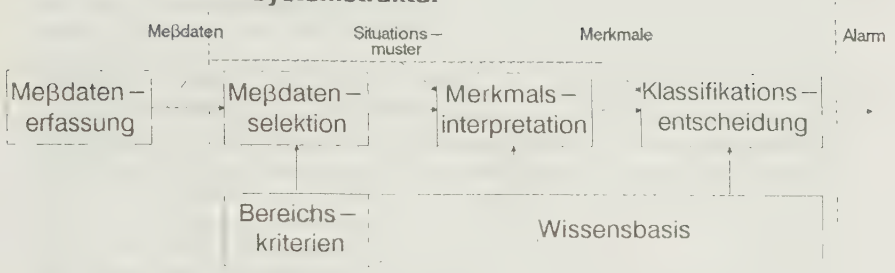


Bild 4.6 : Struktur des heuristisch basierten Überwachungssystems

unterteilt. Ferner soll die separate Interpretation der eindeutigen Generierung von Merkmalen aus dem evtl. mehrdeutigen Situationsmuster dienen. In der Klassifikationsstufe folgt die Entscheidung zum Alarm aufgrund der Merkmale, wobei Rand- und Nebenbedingungen des Prozesses einbezogen werden. Aus der Wahl eines dreistufigen Aufbaus ist zu schließen, daß die Wissensbasis zweckmäßigerweise in mindestens drei Regelgruppen unterteilt wird. Zur Vermeidung langer Inferenzzyklen kann innerhalb der einzelnen Regelgruppen eine verfeinerte Unterteilung in Subgruppen vorgenommen werden.

Durch den in Bild 4.6 dargestellten Aufbau wird berücksichtigt, daß eine möglichst frühzeitige Erkennung kritischer Reaktionszustände gefordert wird. Für den wissensbasierten Entwurf eines Überwachungssystems ist es deshalb erforderlich, daß sich die Meßdatenselektion auf eine möglichst nur wenige Regeln umfassende Regelgruppe der Wissensbasis stützt, um zeitkritische Vorgänge rechtzeitig zu erfassen. Im einfachsten Fall kann ein Kriterium für eine mögliche Gefahrensituation das Überschreiten einer bestimmten Warnschwellen-Temperatur sein. Weiterhin kann z.B. die Tendenz zu beschleunigten Temperaturverläufen durch eine wachsende Differenz zwischen den Meßzeitpunkten berücksichtigt werden. Die Vorverarbeitung der Meßdaten in der Selektionsstufe aufgrund von Regeln, die z.B. derartige Bereichskriterien repräsentieren, würde so lange durchgeführt werden, bis ein die Kriterien erfüllendes Situationsmuster auftritt.

Die Interpretation eindeutiger Merkmale aus dem Situationsmuster erfordert die Berücksichtigung bestimmter Nebenbedingungen. So ist zweckmäßigerweise zunächst festzustellen - vor dem Auslösen evtl. längerer Inferenzzyklen - ob das selektierte Situationsmuster noch vorliegt. Dazu kann z.B. der nächste Meßwert in die Merkmalsextraktion einbezogen werden. Unter der Voraussetzung, daß die Meßdatenselektion auf der Basis des einfachsten Kriteriums, dem Überschreiten einer Grenztemperatur, durchgeführt wird, liefert dieser Folgemeßwert zwar ein Indiz für einen möglicherweise noch kritischen Reaktionsverlauf, andererseits aber kein eindeutiges Merkmal für eine gefährliche Prozeßsituation. Die Tendenz des Temperaturverlaufs kann dann in dieser Stufe z.B. durch den Zeitpunkt der Schwellwertüberschreitung⁸⁾ berücksichtigt werden, wobei Erfahrungswerte über den Prozeß implizit in den entsprechenden Regeln repräsentiert werden. Ferner ist es auch in dieser Stufe denkbar, aufgrund des neu aufgenommenen Meßwerts die Temperaturdifferenzen zwischen den letzten Meßzeitpunkten in die Schlußfolgerung einzubeziehen.

Die in der Interpretationsstufe generierten Merkmale werden, sofern das Selektionsmuster noch vorliegt⁹⁾, in der Klassifikationsstufe zur Alarmentscheidung ausgewertet. Dabei muß, neben den Konditionsteilen der erfüllten Regeln zur Musterinterpretation, die die wissensformulierten Merkmale, z.B. unter Einbeziehung eines Konfidenzfaktors, repräsentieren, erneut die Berücksichtigung von Rand- und Nebenbedingungen in den Inferenzmechanismus einbezogen werden. Explizit auszuwertende Einflußfaktoren können ferner auch in dieser Stufe durch einen weiteren Temperatur-Meßwert und seinen relativen Zeitpunkt gegeben sein.

Mit dem dreistufigen Aufbau des wissensbasierten Überwachungssystems, gestützt auf eine in geeignete Regelgruppen unterteilte Wissensbasis, soll der Anforderung genügt werden, das evtl. zeitkritische Problem der frühzeitigen Erkennung gefährlicher Reaktionszustände hinreichend schnell durchzuführen. Bei der Realisierung des in Abschnitt 4.4 beschriebenen heuristischen Demonstrationssystems wurde die Auswahl der Regelgruppen durch das Setzen globaler Variablen realisiert.

8) Der Zeitpunkt der Grenzwertüberschreitung relativ zum Beginn des Reaktionsprozesses ist in der Regel zu erfassen.

9) Anderenfalls wird mit der erneuten Meßdatenselektion fortgesetzt.

Diese Strategie der Partitionierung wird auch in einigen Shells, z.B. [ACQ86], angewendet, um Konsultationszeiten dadurch zu verkürzen, daß lediglich ein Teil der Wissensbasis in den temporären Arbeitsspeicher geladen wird. Die Auswahl der relevanten Teile der Wissensbasis erfolgt dabei durch eine Kontextsteuerung. Ein anderer Ansatz, der allerdings für Planungsaufgaben entwickelt wurde [ELZ87], benutzt die Separation nicht in dieser statischen Vorgehensweise, sondern dynamisch zur Zerlegung der Problemlösung in die Bearbeitung von Teilaufgaben, bis hin zur elementaren Ausführungsebene. Damit wird ein wesentlicher Vorteil gegenüber älteren Ansätzen erzielt, da kombinatorische Explosionen vermieden werden.

Sind bei der Überwachung des chemische Prozesses mehrere Meßgrößen erfaßbar, bzw. z.B. die Reaktionstemperatur an verschiedenen geometrischen Orten meßbar, so ergibt sich ferner, daß aus der Konstellation der Meßwerte Rückschlüsse über mögliche Falschalarme gezogen werden können. Die wissensbasierte Formulierung des Problems bietet hier die Möglichkeit, durch die - realisierungstechnisch einfache - Ergänzung entsprechender Regeln die Ausschaltung solcher Falschalarme, die z.B. auf Sensorfehler zurückzuführen sind, zu gewährleisten.

4.3.2.3 Systemstruktur unter Integration eines einfachen Tendenzmodells

Die Erkennung kritischer Zustände durch wissensbasierte Überwachungssysteme erfordert eine sorgfältige Separation der Wissensbasis in Regelgruppen zur Vorverarbeitung der Meßdaten und der eigentlichen Merkmalsgenerierung und Alarmerkennung. Das Wissen über komplexere chemische Prozesse, die eine starke Wärmetönung aufweisen, kann bezüglich beschleunigter exothermer Reaktionsvorgänge evtl. nicht so repräsentiert werden, daß eine Verarbeitung im Inferenzprozeß in ausreichender Zeit gewährleistet ist. Sind z.B. die Reaktionsbedingungen durch eine Vielzahl von beeinflussenden Parametern charakterisiert, die zudem stark nichtlineare gegenseitige Einflüsse aufweisen, so ist unter Umständen bereits die Aufstellung geeigneter Bereichskriterien für die Meßdatenselektion - hinsichtlich des Zeitverhaltens derartiger Prozesse - nicht in erforderlichem Maße zu leisten.

Durch die Einbindung einer numerischen Vorverarbeitungsstufe in das Expertensystem, die die Selektions- und Interpretationsstufen des heuristisch basierten Ansatzes zusammenfaßt und ersetzt, kann der Nachteil des nicht-deterministischen Inferenzalgorithmus', die nicht vorhersehbare Suchtiefe, ausgeschaltet und ein bezüglich zeitkritischer Vorgänge verbessertes Verhalten erreicht werden. Dabei muß jedoch berücksichtigt werden, daß aufwendigere algorithmische Lösungsansätze, wie z.B. nichtlineare Filteralgorithmen, vermieden werden, um den realisierungstechnischen Aufwand für das hybride Gesamtsystem in Grenzen zu halten. Ein geeigneterer Ansatz wäre vielmehr ein Tendenzmodell, das auf der Basis des akquirierten Wissens über den betrachteten Prozeß aufgestellt wird, und die groben Merkmale für die Alarmentscheidung liefert. In der Klassifikation der Merkmale können Restriktionen des Modells, die durch Kausalzusammenhänge, Rand- und Nebenbedingungen gegeben sind, berücksichtigt werden. Die Struktur des modellgestützten wissensbasierten Überwachungssystems ist in Bild 4.7 dargestellt.

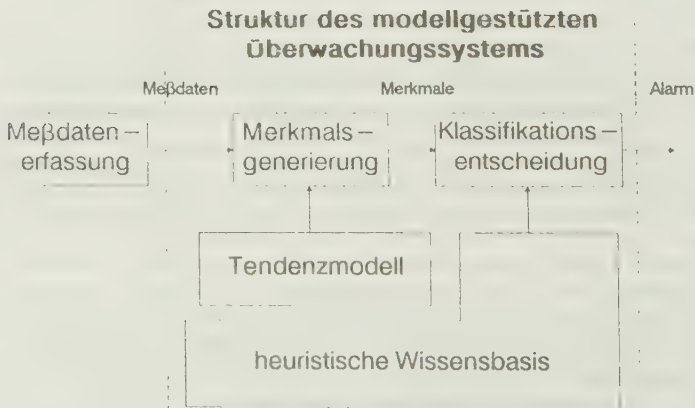


Bild 4.7 : Struktur eines wissensbasierten Überwachungssystems mit integriertem Tendenzmodell

4.3.2.3.1 Auslegung des Tendenzmodells auf der Basis von Proportionalitäts-Zusammenhängen

Betrachtet man die exemplarischen Temperaturverläufe der Polymerisationsreaktion gemäß Bild 4.2, so ist zu bemerken, daß die Reaktionstemperatur, obwohl teils deutliche Spitzen aufweisend, nur bis in begrenzte Bereiche anwächst, weil das Abreagieren der Reaktanden die Selbstbeschleunigung der Reaktion abbremsst. Umgekehrt gilt unterhalb einer minimalen Einfülltemperatur, daß die resultierenden Betriebsbedingungen nicht das Auslösen der Reaktion initiieren. Mit den beiden Grenztemperaturen der Reaktion, die maximal und minimal auftreten können, stehen zwei charakterisierende Parameter zur Verfügung, die in den Entwurf eines Tendenzmodells derart einbezogen werden können, daß sie die Grenzbereiche der Gültigkeit des Modells festlegen. Dabei kann sich die Auslegung dieser Parameter auf Erfahrungswerte aus dem Betrieb der Anlage, die z.B. auch in Meßprotokollen festgehalten sind, stützen.

Um den Bereich kritischer Zustände abzugrenzen, bedarf es zumindest der Kenntnis des Normalbetriebs. Aufgrund dieser Kenntnis, die ebenfalls durch - häufig langjähriges - Betreiben der Anlage vorliegt, lassen sich, unter der Annahme, daß die Reaktionstemperatur als Meßgröße vorliegt und die Reaktionsdauer relativ zum Beginn der Reaktion erfaßbar bzw. abschätzbar ist, zwei weitere Parameter des Tendenzmodells auslegen. Für die relative Reaktionsdauer wird dabei im folgenden vereinfachend angenommen, daß sie durch die normierte dimensionslose Zeit τ_k , betrachtet in den Abtastzeitpunkten $k=1,2,3,\dots,n$, substituiert wird, wobei ferner gelte, daß $\tau_0=0$ ist. Anhand des Verlaufs der "optimalen Normalreaktion", d.h. derjenigen Reaktion, die unter unkritischen Betriebsbedingungen die bestmögliche Produktausbeute ergibt, läßt sich dann zunächst, ähnlich wie beim heuristisch basierten Ansatz, eine Schwellentemperatur $\bar{T}$ festlegen, die den Eintritt des Reaktionsverlaufs in eine mögliche kritische Zone charakterisiert. Als zweiten Parameter, der ein solches Ereignis kennzeichnet, ist es möglich, den Zeitpunkt $\bar{\tau}$ des Überschreitens der Grenztemperatur $\bar{T}$ einzuführen, wobei dieser Zeitpunkt reaktionsspezifisch ausgelegt werden kann, zweckmäßigerweise aber durch das Erreichen der Schwellentemperatur im Fall der Normalreaktion festgelegt wird.

In Analogie zur Betrachtung der Übergangswahrscheinlichkeit in einen Störfall unter Anwendung konventioneller Methoden [MBB77], kann nun ein vereinfachter Ansatz eines Tendenzmodells zur Merkmalsgenerierung auf der Basis der folgenden Regeln aufgestellt werden :

- a) Je höher die Temperatur, desto größer die Wahrscheinlichkeit für das Eintreten eines kritischen Zustandes;
- b) Je schneller die Temperatur steigt, desto eher besteht die Tendenz zu einem gefährlichen Reaktionsverlauf.

(4-25)

Für ein einfaches numerisches Modell sind diese Aussagen so zu interpretieren, daß die Risikowahrscheinlichkeit proportional zur Temperatur und umgekehrt proportional zur relativen Dauer der Reaktion ist. Unter Berücksichtigung der zuvor eingeführten charakteristischen Parameter eines Tendenzmodells zur Merkmalsextraktion wird auf der Basis dieser Regeln die Beziehung

$$R = \bar{\tau}/\tau_k \cdot (1/2 + (T - \bar{T})/(T_{\max} - T_{\min})) , \quad k=1,2,\dots , \quad (4-26)$$

$$\tau_0 = 0 ,$$

vorgeschlagen, wobei berücksichtigt wird, daß sich für das Risikomerkmals R der Grenztemperatur $T = \bar{T}$ zum Zeitpunkt $\bar{\tau}$ der Wert 0.5 ergeben soll.

Um eine breitere Streuung des Wertebereichs des Risikomerkmals R , z.B. auch aus implementierungsspezifischen Gründen, zu erreichen, kann die Beziehung durch einen Faktor mittleren Risikos $\bar{R}$ erweitert werden zu

$$R = \bar{R} \cdot \bar{\tau}/\tau_k \cdot (1 + 2 \cdot (T - \bar{T})/(T_{\max} - T_{\min})) . \quad (4-27)$$

Als Beispiel für eine konkrete Auswahl der Parameter $T_{\max}$, $T_{\min}$, $\bar{T}$ und $\bar{\tau}$ des Tendenzmodells (4-26) sei die mittlere Kurve des Temperaturverlaufs der Polymerisationsreaktion nach Bild 4.2 (3) als optimaler Normalverlauf angenommen. Ein geeignetes Kriterium für die Wahl der Parameter $\bar{T}$ und $\bar{\tau}$ bietet hier der Wendepunkt des Temperaturverlaufs (3), um durch den zu Beginn beschleunigten Reaktionsverlauf zu berücksichtigen, daß eine möglichst hohe Produktausbeute erzielt wird. Nach Überschreiten der Grenztemperatur $\bar{T}$ ist die Tendenz der Temperatursteigerung gegenläufig, was einen unkritischen Reaktionsverlauf erwarten läßt. Die Auslegung der Parameter $T_{\max}$ und $T_{\min}$ kann gemäß den in der Darstellung abgebildeten Verläufen vorgenommen werden. Die Verläufe sind, mit der Kennzeichnung der ausgewählten Parameter, nochmals in Bild 4.8 aufgezeichnet.

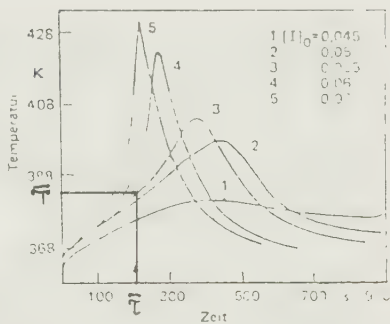


Bild 4.8 : Temperaturverläufe der Polymerisationsreaktion mit Kennzeichnung der Parameter des Tendenzmodells

4.3.2.3.2 Alarmklassifikation auf der Basis der Merkmalsextraktion mit dem Tendenzmodell

Die wissensbasierte Auswertung des Risikomerkmals, ob ein gefährlicher oder ein unkritischer Reaktionsverlauf vorliegt, erfolgt in der Klassifikationsstufe, in der, aufgrund der in der heuristischen Wissensbasis repräsentierten Regeln, eine Alarmentscheidung getroffen wird.

Dabei erscheint als erster Vorteil gegenüber dem rein heuristisch basierten Ansatz, daß die Kenntnis der Parameter des Tendenzmodells und der aktuellen Meßwerte explizit vorliegt und in die Klassifikationsentscheidung einbezogen werden kann. Damit steht ein breiteres Informationsspektrum zur verfeinerten Klassifizierung des Risikomerkmals zur Verfügung, so daß weitere Meßwerte zur Entscheidungsunterstützung nicht aufgenommen

werden müssen, da auch scheinbare Mehrdeutigkeiten des einfachen Extraktionsmodells eliminiert werden können¹⁰⁾. Voraussetzung dafür ist allerdings eine hinreichende Sorgfalt in der Wissensakquirierung, speziell bzgl. solcher Phänomene, die unter ganz bestimmten (extremen) Betriebsbedingungen auftreten können, und eine konsistente Repräsentation des akquirierten Wissens, d.h. vornehmlich die Ausschließung inkausaler Aussagen und Zusammenhänge¹¹⁾.

Ein weiterer Vorteil, betrachtet man die Reaktionsverläufe der Sulfonierung nach Bild 4.3, erscheint darin, daß auch derartige kritische Betriebssituationen, wie das Anspringen einer parasitären Folgereaktion, mit dem modellgestützten Ansatz besser zu erfassen sind, da sich eine Veränderung des normalen, nahezu adiabaten Temperaturverlaufs unmittelbar durch das Risikomerkmals "bemerksam" machen würde¹²⁾.

Der modellgestützte Ansatz erweist sich ferner als einsetzbar für die "inverse" Aufgabe der optimalen Prozeßführung, da durch die Generierung des Risikomerkmals durch das Tendenzmodell auch solche Zustände erfaßt werden, die auf ein unvollständiges Abreagieren der Edukte hinweisen könnten. Entsprechende Maßnahmen aufgrund des rein wissensbasierten Ansatzes zu initiieren, erscheint bezüglich dieses Aspekts realisierungstechnisch wesentlich schwieriger durchzuführen, da mit der Erweiterung des zugehörigen Regelwerks leicht die Grenzen zeitkritischer Probleme erreicht werden können. Dagegen ist die Neu-Parametrisierung des wissensgestützten Tendenzmodells, bzw. die Anpassung der Modellparameter an veränderte Betriebsbedingungen, durch eine (auch off-line vorzunehmende) Veränderung über die Wissensbasis, relativ einfach zu realisieren.

10) Die triviale stationäre Gleichung (4-26) gilt nur in den bestimmten Bereichen des Reaktionsverlaufs, die für die Alarmklassifikation von Bedeutung sind.

11) So ist z.B. im Regelwerk zu berücksichtigen, daß für die exemplarisch betrachtete Klasse von chemischen Prozessen aus kausalen Gründen, Abnahme der Eduktkonzentration etc., eine Reaktion unter entsprechenden Reaktionsbedingungen nicht mehr "durchzündet" kann.

12) Dafür ist u.U. eine Modifizierung des Extraktionsmodells notwendig.

4.3.3 Akquirierung, Strukturierung und Formalisierung des zur Lösung der Überwachungsaufgabe benötigten Wissens

Zur Problematik der Wissensakquisition für die behandelte Aufgabe ist an dieser Stelle eigentlich nichts wesentliches mehr beizutragen, da zum einen, dieser Umstand sei nochmals erwähnt, die notwendigen Experten, die am realen Betrieb einer chemischen Anlage beteiligt sind, im Rahmen dieser Arbeit nicht zur Befragung zur Verfügung standen, und andererseits die dominierenden Aspekte des Wissenserwerbsproblems auf einem allgemeineren Niveau bereits in Abschnitt 3 diskutiert wurden. Desgleichen erübrigen sich Wiederholungen über Wisseninhalte, die sich aus Kausalitäts- und Plausibilitätsbedingungen ergeben, da sie bereits im Zusammenhang mit der Einführung der Grundstruktur des wissensbasierten Überwachungssystems charakterisiert wurden. Es sei hier allerdings nochmals deutlich gemacht, daß die Wissensgewinnung für Domänen derartiger Komplexität das entscheidende zentrale Problem darstellt, und die Übertragung der prinzipiellen Betrachtung der Aufgabe auf kompliziertere Prozesse, wenn auch grundsätzlich möglich, nicht ohne gesteigerten Aufwand durchzuführen ist.

Die Strukturierung des Wissens dient in erster Linie dem Zweck, "nebensächliche" Effekte der Prozeßführung von für die Alarmklassifizierung essentiellen Zusammenhängen zu separieren. Wird ein Entwurf auf Basis der heuristischen Systemstruktur durchgeführt, muß durch das Wissen, das in der Selektionsstufe vorverarbeitet wird, der Tatsache Rechnung getragen werden, daß die Automatisierung des Überwachungsvorgangs durch ein Rechnerprogramm "sturen" Gesetzmäßigkeiten gehorcht.

Beim Anfahrvorgang eines Ammoniakreaktors nach einer Wartungsphase, verhält sich z.B. das Leitwartenpersonal im Sinne der festgelegten "Logik" eines Rechners völlig "unnormale", da akustische und optische Warnsignale scheinbar nicht beachtet und diverse Stelleingriffe, subjektiv betrachtet, willkürlich vorgenommen werden. Da die (in konventioneller Technik ausgeführte) Überwachungsanlage für den kontinuierlichen stationären Betrieb (über Jahre) ausgelegt ist, ist dem Wartenpersonal aber (aus Erfahrung !) bekannt, daß bestimmte (alarmauslösende) Indikatoren in dieser Betriebsphase nicht beachtet werden müssen.

Expertenheuristik wird demnach vorwiegend für die Klassifikation, sowie für die Merkmalsinterpretation von Bedeutung sein, während in der Selektion eher systematisches Wissen repräsentiert wird. Die Situationsklassifikation würde sich für den Anfahrvorgang z.B. durch eine einzige Regel beschreiben lassen, ohne zusätzliche Merkmale (bewußt) zu berücksichtigen, wenn der Adressat der Regel ein neuer anzulernender Kollege ist. Für die Verarbeitung des Vorgangs durch ein Rechnerprogramm wäre es allerdings notwendig, genauere Kenntnis über den Betriebszustand zu erhalten, die bereits in der Selektionsstufe ausgewertet wird.

Grundsätzlich sollte die Wissensstrukturierung der in Abschnitt 4.3.2.2 eingeführten Vorschrift einer Aufteilung in verschiedene Regelgruppen, entsprechend dem mehrstufigen Aufbau des heuristischen Überwachungssystems, gehorchen, um prinzipiell der Forderung nach Realzeitfähigkeit des Systems zu genügen. Unter Berücksichtigung der zuvor genannten Bedingungen kann dann eine Wissensbasis-Struktur erreicht werden, die längere Inferenzzyklen weitgehend ausschließt. Eine Effizienzsteigerung bei der Behandlung komplizierterer Probleme ist ferner möglich, wenn innerhalb der einzelnen Stufen eine verfeinerte Unterteilung der Regelgruppen vorgenommen werden kann.

Für den Ansatz des wissensbasierten Überwachungssystem mit integriertem Tendenzmodell gilt, daß die unmittelbare Klassifizierung der Alarmmerkmale durch die Kenntnis der Parameter des Modells, die der jeweiligen Betriebssituation angepaßt werden können, unterstützt wird, was die Identifizierung der oben beschriebenen Extremsituation der Klassifikationsentscheidung mit Hilfe einer Regel zumindest prinzipiell ermöglichen könnte. Die direkte Auswertung der Meßdaten erleichtert die Strukturierung des Wissens, jedoch gilt auch hier für kompliziertere Prozesse, daß eine Optimierung der Kontrollstrategie durch eine Separierung der Wissensbasis in Regelgruppen, die nach den verschiedenen Reaktionsphasen und Betriebsbedingungen ausgelegt wird, erreicht werden kann, um damit längere Inferenzzyklen bei der Klassifikation der Merkmale zu vermeiden.

Für die formale Darstellung des Wissens erscheinen Produktionsregeln, aufgrund der Ähnlichkeit der Überwachungsaufgabe zu Diagnoseproblemen, am besten geeignet, da die geforderte Auswertung von Meßdaten der vorwärtsverketteten, also datengetriebenen Suchstrategie von Diagnosesystemen entspricht. Ferner ist

durch Produktionsregeln eine geschlossene und konsistente Repräsentation des Wissens zu erreichen, ohne undefinierte "Randzonen" zu schaffen, in denen die fortlaufende Verarbeitung des Wissens abgebrochen wird¹³⁾. Außerdem ist das "Umschalten" zwischen verschiedenen Regelgruppen der Wissensbasis relativ einfach zu realisieren, z.B. durch das Setzen bestimmter globaler Steuervariablen.

Eine Wissensrepräsentation in Form von Frames erscheint, wenn auch aufgrund der kompakten Darstellungsweise vorteilhafter, nicht geeigneter, da durch die uneinheitliche Struktur eine aufwendigere Realisierung der Kontrollstruktur erforderlich wäre, was unter bestimmten Bedingungen die Realzeitfähigkeit des Expertensystems beeinträchtigen würde. Produktionsregeln dagegen erleichtern durch ihre einheitliche Struktur die Realisierung eines simplen, aber effizienten Inferenzmechanismus'.

Andere objektorientierte Repräsentationsformen, wie z.B. semantische Netzwerke, erscheinen grundsätzlich für Diagnoseprobleme ungeeignet, da die Behandlung von Ausnahmen (vgl. Abschnitt 2.1.1.1) eine unvorhersehbare Suchtiefe zur Folge hat, wogegen der prozedurale Charakter von Produktionsregeln eine - zumindest grobe - Abschätzung des Ablaufmechanismus' zuläßt.

Die Wissensdarstellung in Form logischer Beziehungen ist ebenfalls auszuschließen, da die Beschreibung heuristischer Vorschriften und Verhaltensmuster durch z.B. Horn-Klauseln nicht im erforderlichen Maß realisierbar ist, was dagegen in Produktionsregeln durch die Berücksichtigung von Konfidenzfaktoren in die Repräsentation einbezogen werden kann.

13) Ein vollständiges Regelwerk wäre allerdings erst das Ergebnis des erwähnten iterativen Entwicklungsprozesses.

4.4 Realisierung von Demonstrationssystemen zur prinzipiellen Überprüfung des Systementwurfs

Zur grundsätzlichen Verifizierung der Durchführbarkeit der Überwachungsaufgabe mit wissensbasierten Methoden wurden zwei Minimal-Demonstrationssysteme, sowohl für den rein heuristisch basierten Ansatz, als auch für den Systementwurf mit integriertem Tendenzmodell, entwickelt, wobei die Polymerisationsreaktion nach Bild 4.8 als Beispiel verwendet wurde. Für die in Abschnitt 4.4.2 dokumentierten Untersuchungen wurde vorausgesetzt, daß eine Maximaltemperatur von 400 K nicht deutlich überschritten werden sollte, und somit die mittlere Temperaturkurve (3) den zulässigen Grenzfall darstellt. Die Erkennung einer parasitären Zersetzungsreaktion bei der Sulfonierung gemäß Bild 4.3 wurde ebenfalls mit dem Tendenzmodell-Ansatz überprüft, wobei das Erreichen eines Temperaturbereichs größer 130°C im letzten Fünftel der Reaktion als Alarmkriterium gewählt wurde.

4.4.1 Konzipierung der Testsysteme

Da eine umfassende Wissensakquisition aus den schon genannten Gründen nicht durchzuführen war, wurde die Wissensbasis hauptsächlich aus der Kenntnis von Kausalitätsbedingungen und Plausibilitätsannahmen aufgebaut, allerdings stand ein qualifizierter Chemiker (der nicht im Betrieb einer derartigen Anlage tätig ist) zur gelegentlichen Befragung zur Verfügung, was zur Klärung einiger Widersprüche beigetragen hat. Die Wissensrepräsentation erfolgte durch Produktionsregeln. Für die modellgestützte Version des Demonstrationssystems wurde das Tendenzmodell gemäß Gl.(4-27) verwendet.

Beide Systeme wurden vollständig in LISP geschrieben (XLISP1.4 auf AT-kompatiblem PC), wobei auch beim Aufbau der Wissensbasis auf eine besondere Regelsyntax verzichtet wurde, und spezielle LISP-Funktionen genutzt wurden, um der einfachen Struktur der Ablaufsteuerung zu genügen. Als "meßbar" vorausgesetzte Prozeßgrößen wurden Dauer der Reaktion nach - als definiert angenommen - Reaktionsbeginn (in Abtastschritten) und absolute Reaktionstemperatur angenommen, deren Werte aus Datenfiles eingelesen werden.

Der Aufbau der Wissensbasis erfolgte in Form von Produktionsregeln, wobei beim heuristischen Entwurf die vorgeschlagene Separation in 3 Regelgruppen vorgenommen wurde. Es folgen Beispiele für Regeln des heuristisch basierten Demonstrationssystems, wobei je eine Regel jeder Gruppe angegeben wird.

- a) Regel der Selektionsstufe :
 WENN die Reaktionstemperatur höher als ein Grenzwert ist,
 DANN interpretiere das vorliegende Situationsmuster;
- b) Regel der Interpretationsstufe :
 WENN der Zuwachs des neuen Meßwertes größer als eine
 erlaubte Zuwachsrate ist,
 DANN klassifiziere die weitere Tendenz des Merkmals;
- c) Regel der Klassifikationsstufe :
 WENN der aktuelle Zuwachs geringer als der vorherige
 Zuwachs ist,
 DANN interpretiere das neue Situationsmuster.

Bei der programmiertechnischen Realisierung der Produktionsregeln wurden LISP-spezifische Eigenschaften genutzt, um den notwendigen Aufwand zu begrenzen. Die Wissensbasis des heuristischen Systems ist als (quotierte) Liste repräsentiert, die aus Teillisten für jede Regelgruppe besteht, die wiederum Listen der Regeln als Elemente haben. Jede Regelliste ist ein zusammengesetztes Listenpaar aus Bedingungs- und Aktionsteilliste. Zur Vereinfachung besteht jede Bedingungsteilliste aus einer einzigen Unterliste. Die Wissensbasis des Tendenzmodell-Systems ist in gleicher Weise aufgebaut, nur besteht die Wissensbasis hier aus einer einzigen Regelgruppe. Das Tendenzmodell ist ebenfalls als LISP-Funktion realisiert, wobei Gl. (4-27) verwendet wurde, da der gewählte LISP-Dialekt nur die Verarbeitung von Integer-Zahlen erlaubt. Die Parameter des Modells sind als globale Variablen realisiert, wobei sich ihre Auslegung auf Plausibilitätsannahmen stützt. So gilt z.B. für die mögliche auftretende Maximaltemperatur, daß aufgrund des Abreagierens der Ausgangsstoffe nur bestimmte absolute Werte erreicht werden können. In gleicher Weise sind Bedingungen zur Auslegung der anderen Parameter herangezogen worden.

Bei der Konzipierung der Ablaufsteuerung lag der Schwerpunkt in der möglichst einfachen Realisierung, da das Ziel der Simulation lediglich in der Verifizierung des entwickelten Prinzips bestand. Aus dem gleichen Grund wurde auf die Implementation

einer Benutzeroberfläche, einer Akquisitions- und einer komfortablen Erklärungskomponente verzichtet, eine "Tracing"-Einrichtung zur Ausgabe der aktuellen Datenbasis, die schon aus Test-Gründen realisiert wurde, steht aber zur Verfügung.

Die Ablaufsteuerung ist programmiertechnisch durch eine Art Work-Memory-Konzept realisiert. Der aktuelle Inhalt des Work-Memory wird mit den Bedingungsteilen der aktiven Regelgruppe, im Fall des Tendenzmodell-Systems ist dies die gesamte Wissensbasis, verglichen, der Aktionsteil der ersten "feuerbereiten" Regel sodann zur Ausführung gebracht. Ferner regeln zusätzliche globale Steuervariablen den Ablauf der Rekursionen.

4.4.2 Ergebnisse der Simulationen

Die nach einfachen Kriterien ausgelegten Demonstrationssysteme für die Prozeßüberwachung zeigen bei den gewählten Beispielen, daß die Erkennung kritischer Reaktionszustände durch die Anwendung wissensbasierter Methoden in einem frühen Stadium prinzipiell möglich ist. Im Einzelfall, d.h. abhängig von der Auslegung der Kühlung der betrachteten Anlage, kann eine Verfeinerung der Wissensbasis, bzw. für das Überwachungssystem mit integriertem Tendenzmodell eine Modifikation der zu wählenden charakterisierenden Parameter notwendig werden.

Die Darstellung des zeitlichen Verlaufs der "Reaktion" und der aktuelle Diagnosestatus sind bei der Validierung des heuristisch basierten Überwachungssystems durch eine graphische Ausgabe realisiert, die anzeigt, welche Regelgruppe momentan verarbeitet wird. Helle Schraffierung kennzeichnet die Selektions-, dunkle Schraffierung die Interpretationsstufe, der Vollbalken bedeutet, daß Regeln der Klassifikationsstufe verarbeitet werden.

Die Darstellung in Bild 4.9 der "gefährlichen" Temperaturverläufe (Nummern (4) und (5) gemäß Bild 4.8) zeigt, daß eine Meßdatenselektion bis zum Erreichen der gewählten Grenztemperatur (378 K) erfolgt, sodann mit der Interpretation des Situationsmusters fortgesetzt wird, bis aufgrund des Überschreitens

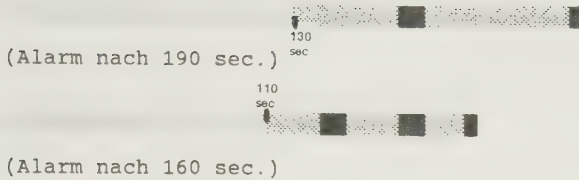


Bild 4.9 : Reaktionsverläufe der Polymerisation,
die zur Alarmauslösung führen (4 und 5);
Heuristisches Demonstrationssystem

eines Alarmzuwachses vor einem bestimmten Grenzzeitpunkt (gewählt $\hat{=}$ 200 sec Reaktionsdauer) der Alarm ausgelöst wird, wobei die vorherige Klassifikation, die durch den Beginn des Durchzündens der Reaktion initiiert wird, noch kein eindeutiges Merkmal zur Alarmauslösung liefert¹⁴⁾.

Für die "zulässigen" unkritischen Verläufe (Bild 4.10) gilt, daß in Bereichen unterhalb der Grenztemperatur die Regeln der Selektionsstufe verarbeitet werden, und oberhalb dieser Schwelle eine Interpretation des Situationsmusters erfolgt, wobei mit dem Abklingen der Reaktion mehrfaches Wechseln zwischen beiden Regelgruppen verbunden ist, da die absolute Reaktions-

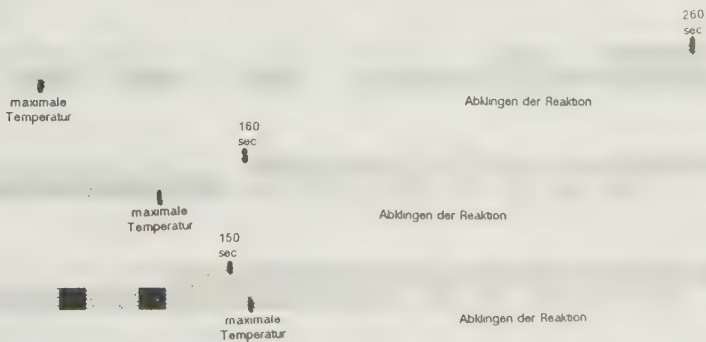


Bild 4.10 : Unkritische Reaktionsverläufe (1, 2 und 3);
Heuristisches Demonstrationssystem

14) Diese Charakteristik ist auch zurückzuführen auf die eingeschränkte Arithmetik des gewählten LISP-Dialekts.

temperatur zwar noch höher als der Schwellwert ist, die Tendenz aber auf das Reaktionsende hinweist. Im Gegensatz zu den Temperaturverläufen (1) und (2) nach Bild 4.8 erreicht die Reaktion gemäß Verlauf (3) Temperaturbereiche, in denen eine Klassifikation des Situationsmusters erfolgt. Dabei liefert das Erreichen von Gebieten negativer Krümmung ein eindeutiges Merkmal für das Ende der Selbstbeschleunigung. Mit dem Abklingen der Reaktion ist dann ferner wieder der mehrfache Wechsel zwischen Selektions- und Interpretationsstufe verbunden.

Da die Wissensbasis des Demonstrationssystems mit integriertem Tendenzmodell aus nur einer einzigen Regelgruppe besteht, ist für die graphische Darstellung der Ergebnisse hier eine andere Notation gewählt. Helle Schraffierung kennzeichnet steigende, dunkle Schraffierung nicht steigende Reaktionstemperaturen.

Bild 4.11 zeigt den Verlauf der Reaktionen (1) und (3) nach Bild 4.8. Zu erkennen ist, daß dem stetigen Anstieg der Reaktionstemperatur bei der Reaktion (1) ein längerer "Verbleib" im Bereich der Maximaltemperatur folgt, d.h. die Reaktion verläuft nahezu adiabat. Der kurze Wechsel zwischen steigender und nicht steigender Temperatur beim zulässigen Grenzfall (3) kennzeichnet den Bereich der Maximaltemperatur, nach deren Überschreiten die Reaktionstemperatur stetig abfällt. In Bild 4.12 ist das stetige Anwachsen der Reaktionstemperatur bis zur Alarmauslösung für die "gefährlichen" Verläufe (4) und (5) dargestellt. Die Auslegung der Parameter des Tendenzmodells erfolgte nach Temperaturverlauf (3) (siehe Bild 4.8), wobei $\bar{T}$ zu 380 K im Zeitpunkt $\bar{\tau} \hat{=} 190$ sec gewählt wurde, was ungefähr dem Wendepunkt in der Krümmung der Kurve entspricht, wobei eine Toleranz von +5 K zur maximal zulässigen Reaktionstemperatur von 400 K berücksichtigt wurde.

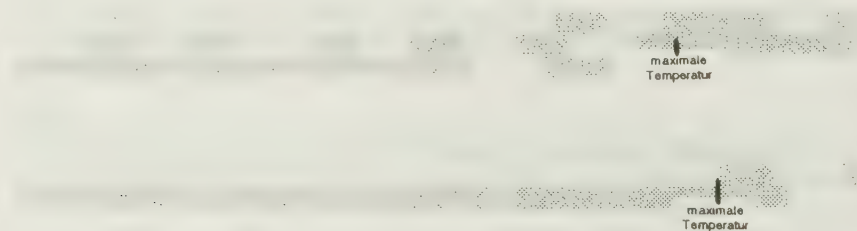


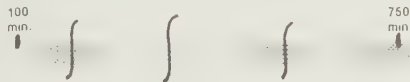
Bild 4.11 : Unkritische Temperaturverläufe der Polymerisationsreaktion (1 und 3); Tendenzmodell-System

(Alarm nach 190 sec.)

(Alarm nach 160 sec.)

Bild 4.12 : Alarmauslösende Reaktionsverläufe (4 und 5)
der Polymerisation; Tendenzmodell-System

Die Erkennung einer parasitären Zersetzungsreaktion mit Hilfe des Tendenzmodell-Systems wurde am Beispiel der Sulfonierungsreaktion gemäß Bild 4.3 verifiziert. Da aufgrund der Einfülltemperatur erst in einem späten Stadium der Reaktion (nach ca. 1000 min) Temperaturbereiche erreicht werden, die die Folgereaktion auslösen, wurde der Grenzzeitparameter $\bar{t}$ zu $\bar{t} \hat{=} 800$ min ausgelegt, wobei die Grenztemperatur $\bar{T}$ oberhalb der Reaktionstemperatur der Normalreaktion zu diesem Zeitpunkt zu $\bar{T} = 130^\circ\text{C}$ gewählt wurde. Die Darstellung in Bild 4.13, die in gleicher Weise gewählt wurde, wie beim modellgestützten Überwachungssystem für die selbstbeschleunigende Reaktion, zeigt, daß nach der Phase des stetigen Temperaturanstiegs zu Beginn die Reaktion im folgenden (verkürzt dargestellten) Zeitraum nahezu adiabatisch verläuft, bis die Reaktionstemperatur in Bereiche gelangt, in denen das Kriterium für die Alarmauslösung, ein Anwachsen des berechneten Risikomerkmals, erfüllt ist. Die relativ frühzeitige Alarmerkennung ist auf die Auslegung der Tendenzmodell-Parameter zurückzuführen, da der exemplarisch gewählte Verlauf z.B. nach ca. 750 min Reaktionsdauer schon eine Reaktionstemperatur von ca. 140°C aufweist.



(Alarm nach 780 min.)

Bild 4.13 : Erkennung einer parasitären Sekundärreaktion
bei der Sulfonierung; Tendenzmodell-System

Die durchgeführten Simulationen zeigen, daß die Erkennung kritischer Reaktionszustände mit Hilfe wissensbasierter Methoden für die gewählten Beispiele prinzipiell möglich ist. Je nach Ausführung der Kühlung der zu betrachtenden Anlage kann allerdings eine andere Auslegung der Parameter, bzw. eine Verfeinerung der Wissensbasis notwendig werden. Die verbleibende Zeitspanne zwischen der Alarmauslösung und dem Erreichen kritischer Temperaturbereiche sollte aber genügen, um Maßnahmen zur Sicherung des Reaktionsapparates zu initiieren und durchzuführen.

5. Zusammenfassung und Ausblick

Die vorliegende Arbeit behandelt, aufbauend auf den informationstheoretischen Grundlagen des Entwurfs von Expertensystemen, die Anwendung wissensbasierter Methoden zur Prozeßüberwachung am Beispiel exothermer chemischer Reaktionen. Mit Hilfe der vorgeschlagenen Verfahren läßt sich das Auftreten kritischer Reaktionszustände in einem so frühen Stadium erkennen, daß Maßnahmen zur Gewährleistung der Sicherheit der Anlage eingeleitet und durchgeführt werden können.

Anhand der in Kapitel 2 durchgeführten Darstellung der bekanntesten Wissensrepräsentationsmethoden und der populärsten Programmierwerkzeuge der Realisierung von Expertensystemen, sowie eines möglichen Entwurfsschemas für wissensbasierte Systeme, wird die Wissensverarbeitungsproblematik im 3. Kapitel aus praxisorientierter Sicht eingeführt und die wesentlichen Aspekte der Umsetzung von Expertenwissen in eine implementierbare formalisierte Darstellung diskutiert.

In Kapitel 4 werden zunächst die exemplarisch betrachteten Reaktionen eingeführt, und gegenüberstellend konventionelle algorithmische Methoden kurz dargestellt, die die Klassifizierung in gefährliche und unkritische Zustände auf der Basis des Kriteriums eines beschleunigten Temperaturanstiegs (4-19) erlauben. Dieses häufig verwendete Kriterium bildet auch die Grundlage der vorgeschlagenen wissensbasierten Verfahren, wobei die regelbasierte Umsetzung (4-25) des Kriteriums genutzt wird, die die Wahrscheinlichkeit eines gefährlichen Zustands so definiert, daß das Verhältnis der Höhe der absoluten Reaktionstemperatur zur momentanen Dauer der Reaktion das Risiko einer kritischen Situation charakterisiert.

Der heuristisch basierte, dreistufige Aufbau des regelbasierten Expertensystems stützt sich auf die direkte Auswertung der Reaktionstemperatur, die als einzige verfügbare Meßgröße vorausgesetzt wird, und die relative Reaktionsdauer nach Reaktionsbeginn, die eine Klassifizierung der Tendenz des Prozeßverlaufs erlaubt.

Im Systementwurf, der die Integration eines Tendenzmodells berücksichtigt, werden die Regeln (4-25) in Form der einfachen stationären Gleichung (4-26) repräsentiert, deren Parameter aus Erfahrungswerten über die "normal" verlaufende Reaktion ausgelegt werden. Ein Ansteigen des Risikomerkmals nach Gleichung (4-27) - bei Überschreitung des mittleren Risikofaktors - kennzeichnet einen gefährlichen Reaktionsverlauf.

Bei entsprechender Auslegung der Tendenzmodell-Parameter erweist sich das modellgestützte Verfahren auch als geeignet für das Erkennen einer parasitären Sekundärreaktion, die die Zersetzung des gewünschten Reaktionsprodukts zur Folge haben könnte.

Die Überprüfung der vorgeschlagenen Lösungsprinzipien wissensbasierter Überwachungssysteme anhand einfacher Beispielreaktionen durch die Realisierung einfacher Demonstrationssysteme zeigt, daß, bei geeigneter Auslegung der Parameter des Tendenzmodells im modellorientierten Ansatz - sowie hinreichender Vollständigkeit der Wissensbasis in beiden Fällen - eine frühzeitige Erkennung gefährlicher Reaktionszustände möglich ist, die es erlaubt, in einer verbleibenden Zeit vor dem Erreichen kritischer Reaktionstemperaturen Maßnahmen zur Sicherung der Anlage zu ergreifen. Die Übertragung der vorgestellten Methoden auf kompliziertere Prozesse erfordert allerdings, bedingt durch den grundsätzlichen Charakter einer wissensbasierten Problemlösung, eine umfangreichere Wissensakquisition, um eine hinreichend vollständige Domänenbeschreibung zu gewinnen. Prinzipiell erscheint es aber möglich, die Erweiterung auf beliebige Prozesse durchzuführen.

Die in dieser Arbeit vorgestellten Verfahren zur Prozeßüberwachung betreffen nur eine Teilaufgabe der Automatisierung technischer Prozesse. Die Führung und Überwachung verfahrenstechnischer Anlagen bietet allerdings auch in anderen Teilgebieten der Gesamtaufgabe die Möglichkeit, wissensbasierte Methoden anzuwenden [AHR87], wobei der Einsatzbereich von Expertensystemen von der Einbettung in umfassendere betriebliche Umgebungen bis zu einem wissensbasierten Gesamtkonzept, einschließlich der

ökonomischen Komponente der Betriebsführung, reichen kann [DAM87]. Im übergeordneten Themenbereich "Diagnose technischer Systeme" sind derzeit einige Aktivitäten anzutreffen [BAT87], die eine zunehmende Integration wissensbasierter Techniken in entsprechende Konzepte von Diagnose-, Überwachungs- und Prozeßleitsystemen erwarten lassen.

Von den dargestellten Lösungsprinzipien für die Überwachungsaufgabe erscheint der modellgestützte Ansatz eines wissensbasierten Überwachungssystems eher geeignet, durch Verfeinerungen und Modifikationen eine verbesserte Prozeßführung zu ermöglichen. So wäre es z.B. möglich, das einfache stationäre Tendenzmodell durch eine qualitative Modellbildung des Prozesses zu substituieren ([KUI87],[OLE87]), um damit auch hinsichtlich Produktausbeute und -qualität eine optimierte Reaktionsführung zu erreichen, ohne komplexere mathematische Ansätze der in den meisten Fällen durch hochgradige Nichtlinearität charakterisierten Prozeßmodelle in die Realisierung einbeziehen zu müssen.

Literatur

- [ACQ86] Lithp Systems BV (eds.) :
User's manual for Acquaint and Acquaint-Light; Lithp Systems BV, 1986
- [AHR86] W. Ahrens :
Stand und Entwicklung technischer Expertensysteme in der Chemie; Fortschritte in der Meß- und Automatisierungstechnik durch Informationstechnik (M.Thoma, G.Schmidt, Hrsg.), Fachberichte Messen, Steuern, Regeln, Nr. 14, Springer, 1986
- [AHR87] W. Ahrens :
Einsatz von Expertensystemen in der Prozeßleittechnik; atp(29), Nr. 10, 1987
- [AIK84] J.S. Aikins, J.C. Kunz, E.H. Shortliffe, R.J. Fallat :
PUFF : An Expert System for Interpretation of Pulmonary Function Data; Readings in Medical Artificial Intelligence : The First Decade (B.C.Clancey, E.H.Shortliffe, eds.), Addison-Wesley, 1984
- [APP85] H.-J. Appelrath :
Von Datenbanken zu Expertensystemen; Informatik-Fachberichte, Nr. 102, Springer, 1985
- [BAR81] A. Barr, P.R. Cohen, E.A. Feigenbaum (eds.) :
The Handbook of Artificial Intelligence, Vol. 1 & 2; William Kaufmann Inc., 1981
- [BAR87] G. Barth :
Moderne Programmierkonzepte; IBM Hochschulkongress Informationsverarbeitung in Lehre und Forschung, Berlin, 1987
- [BAT86] P. Bathelt, M. Kerndlmaier, T. Zink :
Expertensysteme für die technische Diagnose; Fortschritte in der Meß- und Automatisierungstechnik durch Informationstechnik (M.Thoma, G.Schmidt, Hrsg.), Fachberichte Messen, Steuern, Regeln, Nr. 14, Springer, 1986
- [BAT87] P. Bathelt :
Diagnoseexpertensysteme : Stand der Entwicklung im BMFT-Verbundvorhaben TEX-I; GWA, 1987
- [BET84] D. Betz :
XLISP : An Experimental Object Oriented Language; Document File for Version 1.4
- [BLO87] J.A. Blom :
SIMPLEXYS, A Real-Time Expert System Tool; Proc. of the Int. Symp. EXPERT SYSTEMS, Geneva, 1987

- [BRO86] M. Brombacher :
Expertensysteme zur Lastenhefterstellung für die Automatisierung chemischer Verfahren; Fortschritte in der Meß- und Automatisierungstechnik durch Informationstechnik (M.Thoma, G.Schmidt, Hrsg.), Fachberichte Messen, Steuern, Regeln, Nr. 14, Springer, 1986
- [BRW85] L. Brownston, R. Farrell, E. Kant, N. Martin :
Programming Expert Systems in OPS5 - An Introduction to Rule-Based Programming; Addison-Wesley, 1985
- [BUC84] B.G. Buchanan, E.H. Shortliffe :
Rule-Based Expert Systems : The MYCIN Experiments of the Stanford Heuristic Programming Project; Addison-Wesley, 1984
- [CES87] B. Cestnik, N. Lavrac, M. Olave :
The Use of Machine Learning Tools for Building a Management Expert System; Proc. of the Int. Symp. EXPERT SYSTEMS, Geneva, 1987
- [CLO84] W.F. Clocksin, C.S. Mellish :
Programming in Prolog; Springer, 1984
- [COH81] P.R. Cohen, E.A. Feigenbaum (eds.) :
The Handbook of Artificial Intelligence, Vol. 3; William Kaufmann Inc., 1981
- [COL70] A. Colmerauer :
Les Systems-Q ou un Formalisme pour Analyser et Synthetiser des Phrases sur Ordinateur; Department d'Informatique, Univ. de Montreal, 1970
- [DAM87] B. D'Ambrosio, M.R. Fehling, S. Forrest, P. Raulefs, B.M. Wilber :
Real-Time Process Management for Materials Composition in Chemical Manufacturing; IEEE Expert, Summer 1987
- [DAN85] D. Danziger :
Zur Unterdrückung von Temperaturspitzen in polytropen Festbett-Reaktoren durch Katalysatorverdünnung - Simulationen der Ammoniaksynthese und experimentelle Studien der Methanbildung aus Synthesegas; VDI Fortschritt-Berichte, Reihe 3, Nr. 102, 1985
- [DAV82] R. Davis, H. Shrobe, W. Hamscher :
Diagnosis based on description of structure and function; Proc. of the Nat. Conf. on AI, 1982
- [DAV84] R. Davis :
Reasoning from First Principles in Electronic Troubleshooting; Developements in Expert Systems (M.J. Coombs, ed.), Academic Press, 1984
- [DIP85] F. diPrimio, D. Bungers, T. Christaller :
BABYLON als Werkzeug zum Aufbau von Expertensystemen; GI-Kongreß Wissensbasierte Systeme, München, 1985

- [ELZ87] O. El Zubi, R. Münstermann :
Zielgerichtetes Verhalten von Robotern für die Raum-
fahrtanwendungen; 9. Salzhausener Kolloquium der Auto-
matisierungstechnik, 1987
- [EST87] K. Estenfeld :
Prolog-Implementierung - Konzepte und Realisierungen;
Informatik-Spektrum, Nr. 10, 1987
- [EXP86] (Autor ungenannt) :
Expert systems go to work - A survey of tools for
getting started in expert systems; data processing,
vol. 28, no. 3, 1986
- [FRÜ86] H.W. Früchtenicht, T. Wittig :
Ein Ansatz für Echtzeit-Expertensysteme; Fortschritte
in der Meß- und Automatisierungstechnik durch Informa-
tionstechnik (M.Thoma, G.Schmidt, Hrsg.), Fachberichte
Messen, Steuern, Regeln, Nr. 14, Springer, 1986
- [HAR86] P. Harmon, D. King :
Expertensysteme in der Praxis; Oldenbourg, 1986
- [HAY83] F. Hayes-Roth, D.B. Lenat, D.A. Waterman :
Building Expert Systems; Addison-Wesley, 1983
- [HIM78] D.M. Himmelblau :
Fault Detection and Diagnosis in Chemical and Petro-
chemical Processes; Chemical Engineering Monographs,
Vol. 8, Elsevier, 1978
- [HOF87] I. Hofmann, A. Zimmermann :
Expertensysteme für Diagnosezwecke im technischen Be-
reich - Einsatz eines Expertensystems für die Leiter-
plattenprüfung einschließlich exemplarischer Realisie-
rung; Forschungsbericht Nr.2, Universität Bremen, 1987
- [HUB82] L. Hub :
On-Line Überwachung von exothermen chemischen Prozessen
mit dem Oliwa-System; Chem.Ing.-Tech. 54, Nr. 2, 1982
- [IUD84] N.R. Iudica :
Expertensysteme; rtp (26), Nr. 9, 1984
- [KEL86] P. Kelly :
Expert Systems using Shells; data processing, vol. 28,
No. 3, april 1986
- [KOR85] H.-M. Korff :
Expertensysteme zur Diagnose und Konfiguration von Com-
putersystemen; Vorträge zum Workshop Expertensysteme
für die Prozeßautomatisierung, Baden-Baden, GMR-Bericht
7, 1985
- [KUI87] B. Kuipers :
Qualitative Simulation as Causal Explanation; IEEE
Trans. SMC (17), No. 3, May/June 1987

- [LIE86] A. Lieberam :
Expertensysteme für die Verfahrenstechnik; Chem.-Ing.-
Tech. 58, Nr. 1, 1986
- [LIN80] R.K. Lindsay, B.G. Buchanan, E.A. Feigenbaum,
J. Lederberg :
Applications of Artificial Intelligence for Chemical
Inference : The DENDRAL Project; McGraw-Hill, 1980
- [MAC83] B. MacLennan :
Principles of Programming Languages; Holt-Saunders In-
ternational Edition, 1983
- [MBB77] MBB (Hrsg.) :
Technische Zuverlässigkeit, Springer, 1977
- [MCD80] J. McDermott :
The Prolog Phenomenon; Sigart Newsletters 72, 16 - 20
July 1980
- [MER86] P. Mertens, K. Allgeyer, H. Däs, M. Schumann :
Betriebliche Expertensysteme in deutschsprachigen Län-
dern - Versuch einer Bestandsaufnahme; Arbeitsberichte
des Instituts für mathematische Maschinen und Daten-
verarbeitung (Informatik) der Friedrich Alexander Uni-
versität Erlangen Nürnberg, Band 19, Nr. 6, Juli 1986
- [MES85] B. Mescheder :
Funktionen und Arbeitsweise der Expertensystem-Shell
TWAICE; Künstliche Intelligenz und Expertensysteme (S.
Savory, Hrsg.), Oldenbourg, 1985
- [MEY86] U. Meyer :
Konzepte für die Entwicklung von Expertensystemen für
die Fehlerdiagnose technischer Systeme; Tagungsband zum
8. Salzhausener Kolloquium der Automatisierungstechnik
1986 (D. Popovic, Hrsg.), Berichte Elektrotechnik, Re-
port No. 6/87, Univ. Bremen
- [MIL87] R. Milne :
Strategies for Diagnosis; IEEE Trans. SMC (17), No. 3,
May/June 1987
- [MOZ87] I. Mozetic, T. Urbancic :
Hierarchical Qualitative Modeling : A Deep Knowledge
Representation for Efficient Diagnostics; Proc. of the
Int. Symp. EXPERT SYSTEMS, Geneva, 1987
- [NEW72] A. Newell, H. Simon :
Human Problem Solving; Prentice-Hall, 1972
- [NOE85] U. Noelke :
Das Wesen des Knowledge Engineering; Künstliche Intel-
ligenz und Expertensysteme (S. Savory, Hrsg.), Olden-
bourg, 1985

- [OYE87] O. Oyeleye, M.A. Kramer :
Qualitative Simulation of Process Plants; Proc. of the
10th IFAC-World Congress on Automatic Control, 1987
- [PCS85] PCS (Hrsg.) :
Window; PCS-Kundenzeitung, Dezember 1985
- [PEN87] D. Pensak :
Experts Systems - Facts and Fantasies - Implementations
in the Chemical Industry; IBM Hochschulkongress Infor-
mationsverarbeitung in Lehre und Forschung, Berlin,
1987
- [POP87] D. Popovic, U. Meyer, C. Haß, R. Bernhardt-Grisson :
Verbindung von Realzeit- und KI-Sprachen zur Anwendung
in On-Line-Expertensystemen; Proc. of the Int. Symp.
EXPERT SYSTEMS, Genf, 1987
- [PUP86] F. Puppe :
Expertensysteme; Informatik-Spektrum, Nr. 9, 1986
- [QUE84] C. Queinnec :
LISP; Macmillan Computer Science Series, 1984
- [RIE85] P. Rieder, M. Stadel :
Die Programmiersprache Pascal; atp (27), Nr. 9, 1985
- [ROB65] J.A. Robinson :
A Machine-Oriented Logic Based on Resolution Principle;
JACM (23), No. 12, 1965
- [SAV85] S. Savory (Hrsg.) :
Künstliche Intelligenz und Expertensysteme; Oldenbourg,
1985
- [SCH82] H. Schuler :
Frühzeitige Erkennung gefährlicher Reaktionszustände in
chemischen Reaktoren; VDI Fortschritt-Berichte, Reihe
8, Nr. 52, 1982
- [SCH84] H. Schuler :
Frühzeitige Erkennung gefährlicher Reaktionszustände in
chemischen Reaktoren; rt (32), Nr. 6, 1984
- [SCM86] R. Schmidt :
Das Rahmenkonzept als Voraussetzung für den Einsatz von
Prozeßleitsystemen; Fortschritte in der Meß- und Auto-
matisierungstechnik durch Informationstechnik (M.Thoma,
G.Schmidt, Hrsg.), Fachberichte Messen, Steuern,
Regeln, Nr. 14, Springer, 1986
- [SCN86] P. Schnupp :
Wissensbasierte Softwareentwicklung für den Anwendungs-
alltag; GI-Kongreß Wissensbasierte Systeme, 1986

- [SCR77] K.-W. Schrick :
Anwendungen der Kalman-Filter-Technik - Anleitung und
Beispiele; Methoden der Regelungstechnik, Oldenbourg,
1977
- [SHA87] A. Shapour, M. Pecht :
A Coupled Algorithmic-Heuristic Approach for Design
Optimization; IEEE Trans. SMC (17), No. 2, March/April
1987
- [SHI85] J. Shiozaki, H. Matsuyama, E. O'Shima, M. Iri :
An Improved Algorithm for Diagnosis of System Failures
in the Chemical Process; Computers & Chemical
Engineering, Vol. 9, No. 3, 1985
- [TOL87] J.D. Tolman, R. Fell :
An Expert System for 3D Image Understanding; Proc. of
the Int. Symp. EXPERT SYSTEMS, Geneva, 1987
- [VEN86] V. Venkatasubramanian, S.H. Rich :
Reasoning Using Compiled and Deep-Level Knowledge for
Diagnosing Process Plant Failures; KBS for Engineering:
Classification, Education & Control, pp 131-144, 1986
- [WAR77] D.H. Warren :
Implementing Prolog - compiling predicate logic pro-
grams; D.A.I. Research Report, No. 39 u. No. 40, May
1977
- [WAT85] D.A. Waterman :
A Guide to Expert Systems; Addison-Wesley, 1985

Anhang

A Erweitertes Kalman-Filter

Sei das Ergebnis einer mathematischen Modellbildung die nicht-lineare Zustandsdifferentialgleichung

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}) + \underline{G} \cdot \underline{v} \quad , \quad \underline{x}(0) = \underline{x}_0 \quad , \quad (\text{A-1})$$

in der $\underline{f}$ die nichtlineare Funktion mit bekannten Parametern der Zustände $\underline{x}$ und der deterministischen Eingangsgrößen $\underline{u}$ darstellt, und durch den Term $\underline{G} \cdot \underline{v}$ die Einflüsse stochastischer Eingangsgrößen berücksichtigt werden, so lassen sich die unbekannten Zustände $\underline{x}$ des Systems mit Hilfe der Kalman-Filterung rekonstruieren [SCR77].

Unter der Voraussetzung, daß Messungen in den äquidistanten Abtastzeitpunkten $k = 1, 2, 3, \dots$ vorgenommen werden, gilt die diskrete Meßgleichung

$$\underline{z}_k = \underline{h}(\underline{x}_k, k) \underline{x}_k + \underline{w}_k \quad , \quad (\text{A-2})$$

in der $\underline{h}$ eine Funktion der meßbaren Zustände darstellt, und durch die stochastische Größe $\underline{w}$ der Einfluß von Meßstörungen berücksichtigt wird.

Die statistischen Voraussetzungen der Störprozesse $\underline{v}$ und $\underline{w}$ sind, daß sie durch mittelwertfreies Gauß'sches Rauschen zu beschreiben sind, und ferner nicht miteinander korreliert sind

$$E\{\underline{v}(t)\} = \underline{0} \quad , \quad E\{\underline{v}(t) \underline{v}^T(\tau)\} = \underline{Q}(t) \delta(t - \tau) \quad ,$$

$$E\{\underline{w}_k\} = \underline{0} \quad , \quad E\{\underline{w}_k \underline{w}_k^T\} = \underline{R}_k \quad ,$$

$$E\{\underline{v}(t) \underline{w}_k^T\} = \underline{0} \quad . \quad (\text{A-3})$$

Um eine geschlossene Lösung der Filterungsaufgabe zu erzielen, ist es erforderlich, die nichtlinearen Effekte zu linearisieren. Beim linearisierten Kalman-Filter wird dabei von der Kenntnis einer gewünschten Nominaltrajektorie ausgegangen, die

zu linearisierten Differentialgleichungen für die Abweichung von dieser Trajektorie führt. Im Gegensatz dazu werden beim erweiterten Kalman-Filter die Jacobi-Matrizen

$$\underline{F} = (\partial \underline{f} / \partial \underline{x})|_{\underline{x}=\hat{\underline{x}}}$$

$$\underline{H} = (\partial \underline{h} / \partial \underline{x})|_{\underline{x}=\hat{\underline{x}}} \quad (\text{A-4})$$

gebildet, d.h. die Linearisierung entlang des momentanen Schätzwertes vorgenommen, was zu verbesserten Schätzergebnissen führt, wenn die tatsächliche Trajektorie von der nominalen Bewegung deutlich abweicht.

Da die Berechnung des diskreten Äquivalents der Funktion $\underline{f}$ in der Regel nur sehr kompliziert auszuführen ist, wird das erweiterte Kalman-Filter in kontinuierlich-diskreter Mischform ausgelegt, wobei die Prädiktion zwischen zwei Messungen

$$\dot{\hat{\underline{x}}}^* = \underline{f}(\hat{\underline{x}}^*, \underline{u}) \quad \text{und} \quad (\text{A-5})$$

$$\dot{\underline{P}}^* = \underline{F} \underline{P}^* + \underline{P}^* \underline{F}^T + \underline{Q} \underline{Q}^T \quad (\text{A-6})$$

liefert, und durch Up-Date mit der neuen Messung $\underline{z}_k$

$$\hat{\underline{x}}_k = \hat{\underline{x}}_k^* + \underline{K}_k (\underline{z}_k - \underline{H}_k \hat{\underline{x}}_k^*) \quad (\text{A-7})$$

$$\underline{K}_k = \underline{P}_k^* \underline{H}_k^T [\underline{H}_k \underline{P}_k^* \underline{H}_k^T + \underline{R}_k]^{-1} \quad (\text{A-8})$$

$$\underline{P}_k = [\underline{I} - \underline{K}_k \underline{H}_k] \underline{P}_k^* \quad (\text{A-9})$$

der aktuelle Schätzwert $\hat{\underline{x}}_k$, die Kalman-Gain Matrix $\underline{K}_k$ und die Kovarianzmatrix der Schätzfehler $\underline{P}_k$ im Zeitpunkt k bestimmt werden.

Die Auslegung der Matrizen des Filters erfolgt über das Gütekriterium

$$\text{Spur}\{\underline{P}_k\} \rightarrow \text{Min} , \quad (\text{A-10})$$

das zu einer optimalen Bestimmung der Verstärkungsmatrix $\underline{K}_k$ führt.

Der Nachteil des erweiterten Kalman-Filters erweist sich darin, daß, im Gegensatz zur a-priori durchzuführenden Berechnung der Verstärkungsmatrix beim linearisierten Kalman-Filter, durch die

gegenseitige Kopplung von Schätzgleichung (A-7) und Kovarianzgleichung (A-9) der aktuelle Wert der Kalman-Gain-Matrix (A-8) in jedem Abtastzeitpunkt on-line berechnet werden muß.

B Statistische Klassifikationskriterien

Die Überprüfung der Gültigkeit der in den Modellgleichungen (A-1) und (A-2) getroffenen Annahmen wird, bezüglich der Abweichung des Modells vom tatsächlichen Prozeßverlauf [SCH84], anhand der Innovationen

$$y_I(k) = z(k) - z_I(k) \quad \text{und}$$

$$y_{II}(k) = z(k) - z_{II}(k) \quad (\text{B-1})$$

durchgeführt, wobei der Index I das Prozeßmodell charakterisiert, mit dem der bestimmungsgemäße Normalbetrieb beschrieben wird, und der Index II die Berücksichtigung eines Störfalls, z.B. das Anspringen einer parasitären Sekundärreaktion, im parallelen zweiten Modell kennzeichnet.

Unter der Annahme, daß die Modellannahmen zutreffend sind, genügen die Innovationen $y_I(k)$ und $y_{II}(k)$ der Beschreibung durch normalverteilte weiße Zufallsfolgen mit bekannten statistischen Eigenschaften für die die Wahrscheinlichkeitsdichtefunktionen

$$p_I(y_I) = [(2\pi)^m \det S_I(k)]^{-1/2} \cdot e^{-1/2 y_I^T(k) S_I^{-1}(k) y_I(k)}$$

$$p_{II}(y_{II}) = [(2\pi)^m \det S_{II}(k)]^{-1/2} \cdot e^{-1/2 y_{II}^T(k) S_{II}^{-1}(k) y_{II}(k)} \quad (\text{B-2})$$

zu berechnen sind. Die Berechnung der bedingten a-posteriori-Wahrscheinlichkeiten der Klassenzugehörigkeit, d.h. ob sich der

Meßwert $\underline{z}$ in einem Gebiet gefährlicher oder kritischer Zustände befindet, läßt sich durch Integration der Wahrscheinlichkeitsdichte über das Gebiet des "normalen" Zustands

$$\Pr\{\underline{x} \in \Omega_0 | \underline{z}\} = \iint_{\Omega_0} p(\underline{x}') d\underline{x}' \quad (\text{B-3})$$

berechnen, woraus dann letztlich mit

$$\Pr\{\Omega_0 | \underline{z}\} = p_I(\chi_I) / p_{II}(\chi_{II}) \Pr\{\Omega_0\} \quad (\text{B-4})$$

ein Kriterium für die Zugehörigkeit zum Gebiet unkritischer Reaktionszustände resultiert.

Lebenslauf

Uwe Meyer, geb. am 11.5.1958 in Bremen

Eltern : Johann Meyer, Rentner
Gertraude Meyer, geb. Beinlich,
Rentnerin

Schulbildung :	1965 - 1968	Grundschule Karl-Lerbs-Straße, Bremen
	1968 - 1977	Gymnasium Huckelriede, Bremen
	1977/78	15-monatiger Grundwehrdienst

Studium : Herbst 1978 Aufnahme des Studiums der
 Elektrotechnik an der Universi-
 tät Bremen

Herbst 1980 Wahl der Studienfachrichtung
Automatisierungstechnik

30.10.1984 Tag der mündl. Diplomprüfung

Absolvierung von 3 Monaten Fach-
praktikum während des Studiums

Berufs-
tätigkeit : 1.1.1985 -
31.12.1987

Wissenschaftlicher Mitarbeiter
am Institut für Automatisie-
rungstechnik der Universität
Bremen

seit 1.5.1988 Wissenschaftlicher Mitarbeiter
am Institut für Angewandte und
Physikalische Chemie

FOTODRUCK J. MAINZ GmbH
Neupforte 13, Nähe Markt, D-5100 Aachen
Telefon: (0241) 27305 und (0241) 38606